(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5047988号
(P5047988)

(45)発行日　平成24年10月10日(2012.10.10)　(24)登録日　平成24年7月27日(2012.7.27)

(51)Int.Cl.　　F I
G06F 12/00 (2006.01)　　G06F 12/00 531D
G06F 3/06 (2006.01)　　G06F 3/06 301F
　　G06F 12/00 545A

請求項の数 20　(全 89 頁)

(21)出願番号　特願2008-548792 (P2008-548792)
(86)(22)出願日　平成18年11月30日 (2006.11.30)
(65)公表番号　特表2009-522659 (P2009-522659A)
(43)公表日　平成21年6月11日 (2009.6.11)
(86)国際出願番号　PCT/US2006/061431
(87)国際公開番号　WO2008/069811
(87)国際公開日　平成20年6月12日 (2008.6.12)
審査請求日　平成21年11月30日 (2009.11.30)
(31)優先権主張番号　60/754,726
(32)優先日　平成17年12月29日 (2005.12.29)
(33)優先権主張国　米国 (US)
(31)優先権主張番号　11/371,304
(32)優先日　平成18年3月8日 (2006.3.8)
(33)優先権主張国　米国 (US)

(73)特許権者　507303550
アマゾン・テクノロジーズ・インコーポレーテッド
アメリカ合衆国・８９５０７・ネバダ州・レノ・ピーオーボックス　８１０２
(74)代理人　100064621
弁理士　山川　政樹
(74)代理人　100098394
弁理士　山川　茂樹
(72)発明者　バーミューレン，アラン・エイチ
アメリカ合衆国・９８１４４・ワシントン州・シアトル・１２ティエイチ　アベニュ　サウス・１２００・スイート　１２００

最終頁に続く

(54)【発明の名称】ウェブサービスクライアントインターフェースを有する分散型ストレージシステム

(57)【特許請求の範囲】
【請求項１】
システムメモリに接続した１又は複数のプロセッサを備えたデータストレージ・ウェブサービス用のコンピュータにより実行可能な方法であって、当該方法は、
クライアントコンピュータに対してデータストレージ・ウェブサービスを提供するステップであって、
前記データストレージ・ウェブサービス用のコンピュータは、
ウェブサービス要求を通じて前記クライアントコンピュータに対して適用可能であるデータストレージのウェブサービス操作を定義するウェブサービスアプリケーションプログラミングインターフェース（ＡＰＩ）を実装し、
１つ以上の前記ウェブサービス操作を特定する前記ウェブサービスＡＰＩに従ってフォーマットされたウェブサービス要求を受信するために、インターネット系のアプリケーション層データ転送プロトコルに従ってアドレス可能であり、
データオブジェクトを格納するための前記ウェブサービスＡＰＩに従ってフォーマットされたウェブサービス要求を受信するのに応答して、前記クライアントコンピュータから供給されたデータオブジェクトを格納する、ステップと、
前記インターネット系のアプリケーション層データ転送プロトコルに従って、データオブジェクトにアクセスするためのクライアント要求を示す前記ウェブサービスＡＰＩに従ってフォーマットされたウェブサービス要求を受信するステップであって、前記データオブジェクトの所定の１つにアクセスするための前記クライアント要求の所定の１つは、前

記所定データオブジェクトに対応するキー値を含む、ステップと、
複数のストレージノード上に前記データオブジェクトのコピーを保存するステップであって、前記コピーのそれぞれは、それぞれのロケータ値を介してアクセス可能であり、前記ロケータ値のそれぞれは、前記データストレージ・ウェブサービス内で固有である、ステップと、
前記データオブジェクトのそれぞれに対するそれぞれのキーマップエントリを保存するステップであって、前記任意データオブジェクトについては、前記それぞれのキーマップエントリは、前記クライアント特定キー値と、前記所定のデータオブジェクトのそれぞれの保存されたコピーに対応するそれぞれのロケータ値を含む、ステップと、
前記任意クライアント要求の受信に応じて、前記キー値に対応する１つまたは複数のロケータ値を識別するために、前記それぞれのキーマップエントリにアクセスするステップであって、前記１つまたは複数のロケータ値の特定の１つについては、対応するストレージノードにアクセスし、対応するコピーを取り出すステップと、
を備えることを特徴とする方法。
【請求項２】
前記対応するコピーを取り出す前に、さらに前記所定のクライアント要求が、前記所定のデータオブジェクトへアクセスするのに十分な特権を有するかどうかを判断するステップと、前記所定のクライアント要求が十分な権利が与えられていない場合には、前記所定のクライアント要求を拒否するステップと、をさらに備えることを特徴とする請求項１に記載の方法。
【請求項３】
前記インターネット系のアプリケーション層データ転送プロトコルに従って、データオブジェクトを保存するために前記ウェブサービスＡＰＩに従ってフォーマットされたクライアント要求を受信するステップをさらに備え、前記データオブジェクトの特定の１つを保存するための前記クライアント要求の特定の１つは、前記特定データに対応するキー値を含むことを特徴とする請求項１または２に記載の方法。
【請求項４】
前記特定データオブジェクトを保存するための料金を判断するステップをさらに備えたことを特徴とする請求項３に記載の方法。
【請求項５】
前記特定のクライアント要求に応じて、前記特定のデータオブジェクトの１つ以上のコピーを１つまたは複数の対応するストレージノードに保存するステップと、
前記特定のデータオブジェクトの所定のコピーの保存に応じて、前記所定のコピーに対応するロケータ値を受信ステップと、をさらに備えることを特徴とする請求項３に記載の方法。
【請求項６】
前記特定のデータオブジェクトの前記１つまたは複数のコピーが保存される、前記１つまたは複数の対応するストレージノードを、ストレージ規定に従って選択するステップをさらに備えたことを特徴とする請求項４に記載の方法。
【請求項７】
前記ストレージ規定は、前記特定のデータオブジェクトを保存するための前記特定のクライアント要求が完了することを示す前に、対応するストレージノードへ永久に保存されたことを示すために必要となる多くのコピーを特定することを特徴とする請求項６に記載の方法。
【請求項８】
前記ストレージ規定は、生成される前記特定のデータオブジェクトの所望のコピーの数をさらに特定することを特徴とする請求項７に記載の方法。
【請求項９】
前記キーマップエントリの所定の１つに対して、前記所定のキーマップエントリのそれぞれのロケータ値に対応する各コピーが、アクセス可能かどうかの判断をするために、前

記それぞれのキーマップエントリを調査するステップをさらに備えることを特徴とする請求項８に記載の方法。

【請求項１０】

前記所定のキーマップエントリについて、前記所定のキーマップエントリのそれぞれのロケータ値に対応するアクセス可能なコピーの数が、前記コピーの所望の数よりも少ない場合、前記コピーの所望の数を満たすために十分である付加的なコピーを作成するステップをさらに備えることを特徴とする請求項９に記載の方法。

【請求項１１】

前記複数のストレージノードは複数の領域に分散され、前記ストレージ規定は、前記特定のデータオブジェクトを保存するための前記特定のクライアント要求が完了することを示す前に、前記１つまたは複数のコピーが永久に保存されることを示すことが必要となる最少数の領域を特定することを特徴とする請求項６に記載の方法。

【請求項１２】

前記複数の領域の任意の２つの領域間において、ストレージノードの不具合の可能性の相関関係がしきい値未満であることを特徴とする請求項１１に記載の方法。

【請求項１３】

前記ストレージ規定は、可能な場合、前記１つまたは複数のコピーのうちの少なくとも１つが、前記領域の所定のものに位置するストレージノードに書き込まれることをさらに特定することを特徴とする請求項１１に記載の方法。

【請求項１４】

前記データストレージ・ウェブサービス用のコンピュータによって受信された前記所定のデータオブジェクトへのアクセスのための前記クライアント要求のうちの特定の１つは、前記所定のデータオブジェクトの特定のコピーに対応する特定のロケータ値を含むことを特徴とする請求項１１に記載の方法。

【請求項１５】

アクセスのための前記特定のクライアント要求の受信に応じて、前記キーマップエントリにアクセスすることなく、前記特定のロケータ値を介し、対応するストレージノードから前記特定のコピーを取り出すステップをさらに備えることを特徴とする請求項１４に記載の方法。

【請求項１６】

前記インターネット系のアプリケーション層データ転送プロトコルは、ＲｅｐｒｅｓｅｎｔａｔｉｏｎａｌＳｔａｔｅＴｒａｎｓｆｅｒ（ＲＥＳＴ）ウェブサービスモデルを実行し、前記インターネット系のアプリケーション層データ転送プロトコルに従って前記所定のクライアント要求を受信することは、ＨｙｐｅｒｔｅｘｔＴｒａｎｓｆｅｒＰｒｏｔｏｃｏｌ（ＨＴＴＰ）のバージョンに従いフォーマット化された要求を受信することを含むことを特徴とする請求項１に記載の方法。

【請求項１７】

前記インターネット系のアプリケーション層データ転送プロトコルは、文書ベースウェブサービスモデルを実行し、

前記インターネット系のアプリケーション層データ転送プロトコルに従って前記所定のクライアント要求を受信することは、ＳｉｍｐｌｅＯｂｊｅｃｔＡｃｃｅｓｓＰｒｏｔｏｃｏｌ（ＳＯＡＰ）のバージョンに従い、カプセル化した文書の受信をすることを含み、前記所定のクライアント要求の内容は、前記文書に含まれ、ＸＭＬのバージョンに従いフォーマットされることを特徴とする請求項１に記載の方法。

【請求項１８】

前記データオブジェクトのそれぞれに対するそれぞれのキーマップエントリを保存するステップは、階層的に配置された複数のインデックスノードを含み、それぞれが関連するタグ値を有するインデックスデータ構造内に保存されたキーマップエントリを索引付けするステップを含み、

前記キーマップエントリのそれぞれは、インデックスノードのそれぞれの１つと対応し

、所定の対応するインデックスノードを有する前記所定のキーマップエントリについては、前記所定の対応するインデックスノードの各上位と関連する各タグ値が、前記所定のキー値のプレフィックスであることを特徴とする請求項１に記載の方法。

【請求項１９】

前記請求項１乃至１８のいずれかに記載の方法を実施するように構成された複数のコンピュータより構成されるシステム。

【請求項２０】

インストラクションを備えるコンピュータにアクセス可能な媒体であって、前記インストラクションは、データストレージ・ウェブサービス用のコンピュータにより実行され、

クライアントコンピュータに対してデータストレージ・ウェブサービスを提供するステップであって、

前記データストレージ・ウェブサービス用のコンピュータは、

ウェブサービス要求を通じて前記クライアントコンピュータに対して適用可能であるデータストレージのウェブサービス操作を定義するウェブサービスアプリケーションプログラミングインターフェース（ＡＰＩ）を実装し、

１つ以上の前記ウェブサービス操作を特定する前記ウェブサービスＡＰＩに従ってフォーマットされたウェブサービス要求を受信するために、インターネット系のアプリケーション層データ転送プロトコルに従ってアドレス可能であり、

データオブジェクトを格納するための前記ウェブサービスＡＰＩに従ってフォーマットされたウェブサービス要求を受信するのに応答して、前記クライアントコンピュータから供給されたデータオブジェクトを格納する、ステップと、

データオブジェクトへアクセスするためのクライアント要求を示す前記ウェブサービスＡＰＩに従ってフォーマットされたウェブサービス要求を処理するステップであって、前記データオブジェクトにアクセスするための前記クライアント要求は、前記インターネット系のアプリケーション層データ転送プロトコルに従って受信され、前記データオブジェクトの所定の１つへアクセスするための前記クライアント要求の所定の１つは、前記所定のデータオブジェクトに対応するキー値を含む、ステップと、

複数のストレージノード上に前記データオブジェクトのコピーを保存することを指示するステップであって前記ロケータ値のそれぞれは、前記データストレージ・ウェブサービス内で固有であるステップと、

前記データオブジェクトのそれぞれに対するそれぞれのキーマップエントリを保存することを指示するステップであって、前記所定のデータオブジェクトについては、前記それぞれのキーマップエントリは、前記クライアント特定キー値と、前記所定のデータオブジェクトのそれぞれの保存されたコピーに対応するそれぞれのロケータ値とを含むステップと、

を実行することが可能であり、

前記所定のクライアント要求を処理するステップは、前記キー値に対応する１つまたは複数のロケータ値を識別するための前記それぞれのキーマップエントリへアクセスするステップを含み、前記１つまたは複数の特定の１つについては、対応するストレージノードにアクセスし、対応するコピーを取り出すステップを含むことを特徴とするインストラクションを備えることを特徴とするコンピュータにアクセス可能な媒体。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明はデータストレージシステム、より具体的には、ウェブサービスとしてストレージへのアクセスを提供するように構成されるストレージシステムに関連する。

【背景技術】

【０００２】

多くの異なるコンピュータアプリケーションは、様々な種類のアプリケーションデータの持続的保存のために、あるストレージ媒体に依存する。例えば、一般的なオフィスアプ

リケーションおよびマルチメディアアプリケーションは、中でも、文書、集計表、静止画像、音声及びビデオデータなどの様々な種類やフォーマットのアプリケーションデータを作成し、使用する。このようなデータは頻繁に、ユーザが繰り返しアクセス又は使用できるように保存される。例えば、ユーザは、一定期間にわたり多くの文書又は他のデータを保存し、作業することを希望する場合があり、必要な時に予想できる状態で、データを速やかに使用できることを望む場合がある。

【０００３】

従来のコンピュータ関係システムにおいて、持続的なアプリケーションデータの保存のためにアプリケーションによって使用されるストレージ媒体は、光学又は半導体ストレージデバイスも使用されるが、最も一般的には磁気固定ドライブ又は「ハードドライブ」である。このようなデバイスは、アプリケーションを実行するコンピュータシステム内に統合されているか、あるいはローカルの周辺インターフェース又はネットワークを介してそのシステムにアクセス可能である。一般的に、アプリケーションストレージとしての役割をするデバイスは、ファイルシステムインターフェースなど、ストレージアクセスを必要とする様々なアプリケーションへの一貫したストレージインターフェースを提供するためのデバイスレベルでの動作を管理する、オペレーティングシステムによって管理される。

【０００４】

この従来型モデルのアプリケーションストレージには、いくつかの限界が存在する。第１に、概して従来型モデルは、アプリケーションデータのアクセス性を制限する。例えば、アプリケーションデータが、特定のコンピュータシステムのローカルハードドライブに保存される場合、他のシステムで実行されているアプリケーションからアクセス不可能である。データがネットワークでアクセス可能なデバイスに保存される場合でも、周辺ネットワーク外のシステムで実行するアプリケーションは、そのデバイスにアクセスすることが不可能である場合がある。例えば、セキュリティ上の理由で、企業の外部のシステムが企業内のシステム又はリソース情報にアクセスできないように、企業は一般にローカルエリアネットワーク（ＬＡＮ）へのアクセスを制限する。したがって、携帯デバイス上（例えば、ノート型又は手持ちサイズコンピュータ、個人情報端末、携帯電話デバイスなど）で実行するアプリケーションは、固定されたシステム又はネットワークに持続的に関連するデータへのアクセスに困難となる場合がある。

【０００５】

また従来のアプリケーションストレージモデルも、保存されたデータの信頼性を十分に確認することができない場合がある。例えば、従来のオペレーティングシステムは、一般的に初期設定で１つのストレージデバイス上に１つのアプリケーションデータのコピーを保存し、データの冗長性が望まれる場合に、ユーザ又はアプリケーションに自らのアプリケーションデータのコピーを作成し管理することを要求する。個々のストレージデバイス又は第三者のソフトウェアがある程度の冗長性を提供する場合もあるが、アプリケーションが利用可能なストレージリソースがアプリケーション装置よって大いに異なるため、これらの特徴はアプリケーションにとって一貫して利用可能ではない場合がある。またオペレーティングシステム介在の従来のストレージモデルも、プラットフォームを越えるデータへのアクセス性を制限してもよい。例えば、異なるオペレーティングシステムは、異なる、互換性のないフォーマットで同一アプリケーション用のデータを保存する場合があり、１つのプラットフォーム上（例えばオペレーティングシステム及び基礎をなすコンピュータシステムハードウェア）で実行しているアプリケーションのユーザが、異なるプラットフォーム上で実行しているアプリケーションによって保存されたデータにアクセスすることが困難である場合がある。

【発明の開示】

【課題を解決するための手段】

【０００６】

分散型、ウェブサービス系のストレージシステムの様々な実施態様を開示する。一実施態様に従い、システムは、ウェブサービスプロトコルに従い、データオブジェクトにアク

セスするためのクライアント要求を受け取るように構成されるウェブサービスインターフェースを含んでいる。所定のデータオブジェクトへのアクセスのための所定のクライアント要求は、所定のデータオブジェクトと対応するキー値を含んでいる。また該システムは、データオブジェクトの複製を保存するように構成される多くのストレージノードを含み、各複製はそれぞれのロケータ値を介しアクセス可能であり、ロケータ値はそれぞれシステム内で固有である。該システムは、データオブジェクトのそれぞれに対するそれぞれのキーマップエントリを保存するように構成されるキーマップインスタンスをさらに含んでもよく、それぞれのキーマップエントリは、所定のデータオブジェクトのそれぞれの保存された複製に対応するキー値及び各ロケータ値を含む。また該システムは、ウェブサービスインターフェースからデータオブジェクトにアクセスするためのクライアント要求を受け取るように構成されるコーディネータをさらに含んでもよい。所定のクライアント要求に応じて、コーディネータはキー値に対応する１つ以上のロケータ値を認識するためにキーマップインスタンスにアクセスするように構成され、特定のロケータ値に対しては、対応する複製を取り出すために対応するストレージノードにアクセスするように設定されてもよい。

【０００７】

該システムの特定の実装において、該ウェブサービスインターフェースは、ウェブサービスプロトコルに従い、データオブジェクトを保存するためのクライアント要求を受け取るようにさらに設定されてもよく、特定のデータオブジェクトを保存するための特定のクライアント要求は本特定データオブジェクトと対応するキー値を含む。該コーディネータは、ウェブサービスインターフェースからデータオブジェクトを保存するためのクライアント要求を受け取るようにさらに設定されてもよく、特定のクライアント要求に応じて、該コーディネータは１つ以上の対応するストレージノードへ特定のデータオブジェクトの１つ以上の複製を保存するように設定されてもよい。特定のデータオブジェクトの所定の複製を保存することに応じて、所定のストレージノードは、コーディネータに、所定の複製に対応するロケータ値を返すように設定されてもよい。

【０００８】

本発明は、種々の修正及び代替形態が可能であるが、その特定の実施形態は図面によって一例として示され、本願に詳述される。しかしながら、図面及び詳細な説明は、本発明に開示される特定の形態に制限することを意図せず、逆に、添付の請求項によって定義される本発明の精神及び範囲内に含まれるすべての修正、同等物、及び代替手段を扱うことを理解されたい。

【発明を実施するための最良の形態】

【０００９】

序文

コンピュータアプリケーションが地理的に分散されるばかりでなく、さらにデータ集中的になるにつれ、アプリケーションデータへの信頼性のある、位置独立型のアクセスの必要性が増加する。例えば、オーサリング、ストレージ、再生アプリケーションなどのマルチメディアアプリケーションは、マルチメディアコンテンツの質と量が向上するにつれ、データストレージ量の増大を必要とする。さらに、データを保存するデバイスの位置に関係ない様々な場所からアプリケーションデータにアクセスすることが望ましい場合がある。例えば、多くのコンピュータがディスク系の相当な量のストレージを含むが、これらのストレージに一貫した便利な方法で遠隔からアクセスすることは、技術上及び安全上の困難をもたらす。

【００１０】

個々のコンピュータを、単独で個々の内部ストレージリソース又はプロビジョニングローカルネットワーク系のストレージリソース（例えば、ネットワーク接続ストレージ（ＮＡＳ）、ストレージ・エリア・ネットワーク（ＳＡＮ）など）に依存するように構成されることとは対照的に、インターネット接続データストレージサービスは、例えば、ウェブサービス（ＷＳ）プロトコルなどのインターネット系のプロトコルを介し、クライアント

に包括的なストレージサービスを提供するように設定されてもよい。ウェブサービスプロトコルなどのインターネット系プロトコルは、一般的に基礎をなすソフトウェア又はハードウェア上で独立して機能するため、一般的にプラットフォーム独立型である。したがって、ウェブサービスとしてデータストレージ機能を提供することは、アプリケーションのホストシステム又はローカルネットワークに実装されるストレージリソースから独立した任意の大きさのストレージへ多くの異なる種類のアプリケーションが直接アクセスすることを可能にする可能性がある。さらに、ウェブサービスアクセス可能ストレージは、一般的にインターネットアクセスを提供する任意の場所からでもアクセス可能であってもよい。ウェブサービスアクセス可能ストレージは、異なるデバイス又はアプリケーションによる共有データへのリモートアクセス、実行中の個々のアプリケーションによる幅広く分散されたデータへのリモートアクセス、共同で作業する分散したユーザ間のデータへのアクセス及びデータの共有、分散したユーザへのアプリケーション結果データの配布、及び多くのその他の同様の機能などの多数の異なるコンピュータ機能の実装を容易にし得る。

【００１１】

以下の論考において、ウェブサービス系ストレージシステムにおいて使用されてもよい、可能なデータストレージモデルの一実施形態を説明する。続いて、データストレージモデルに従いストレージサービスを提供するように構成されてもよいストレージサービスシステムを開示し、その様々な要素を詳しく説明する。

【００１２】

ストレージサービスユーザインターフェース及びストレージモデルの概略

ウェブサービスなどの、サービスとしてのデータストレージをユーザに提供するためのストレージモデルの一実施形態を図１に示す。図示されたモデルにおいて、ストレージサービスインターフェース１０は、ストレージサービスに対する対顧客又は対ユーザインターフェースとして提供される。インターフェース１０によってユーザに提示されるモデルに従い、ストレージサービスは、インターフェース１０を介して任意の数のバケット２０ａ－ｎとして体系化されてもよい。各バケット２０は、任意の数のオブジェクト３０ａ－ｎを保存するように設定されてもよく、ストレージサービスのユーザによって特定されるデータを入れ替わりに保存してもよい。

【００１３】

以下にさらに詳しく述べるように、一部の実施形態において、ストレージサービスインターフェース１０は、ウェブサービスモデルに従い、ストレージサービスとそのユーザ間の情報のやり取りをサポートするように設定されてもよい。例えば、一実施形態において、インターフェース１０は、例えばｈｔｔｐ：／／ｓｔｏｒａｇｅｓｅｒｖｉｃｅ．ｄｏｍａｉｎ．ｃｏｍなどのＵｎｉｆｏｒｍ Ｒｅｓｏｕｒｃｅ Ｌｏｃａｔｏｒ（ＵＲＬ）を有するウェブサービスエンドポイントとして、クライアントによってアクセス可能であってもよく、サービスクライアントによって作成されたウェブサービスコールは処理対象となる場合がある。一般的に言えば、ウェブサービスは、Ｈｙｐｅｒｔｅｘｔ Ｔｒａｎｓｐｏｒｔ Ｐｒｏｔｏｃｏｌ（ＨＴＴＰ）又は他の適切なプロトコルのバージョンなどの、１つ以上のインターネット系のアプリケーション層データ転送プロトコルを含む要求インターフェースを介し、要求するクライアントが使用できるように作られた、あらゆる種類のコンピュータサービスを示すことができる。

【００１４】

ウェブサービスは、様々な許可サービスプロトコルを使用し、様々なアーキテクチャ形式で実装されてもよい。例えば、Ｒｅｐｒｅｓｅｎｔａｔｉｏｎａｌ Ｓｔａｔｅ Ｔｒａｎｓｆｅｒ（ＲＥＳＴ）形式のウェブサービスアーキテクチャにおいて、ウェブサービスコールに関するパラメータ（例えば、要求されるサービスの種類を特定すること、ユーザ資格、操作すべきユーザデータ、など）は、ＨＴＴＰ ＧＥＴ又はＰＵＴ命令など、ウェブサービスエンドポイントに対するウェブサービスコールを呼び出すデータ転送命令に対するパラメータとして特定されてもよい。一部の実装において、各ウェブサービスコールが、外部状態情報を参照することなくそのコールを処理するために必要なすべての情報

を含んでいる場合があるため、ＲＥＳＴ形式ウェブサービスアーキテクチャは処理状態を把握しない。ＲＥＳＴ形式ウェブサービスアーキテクチャとは対照的に、文書系又はメッセージ系ウェブサービスアーキテクチャは、ウェブサービスエンドポイントに送信され、解読され、エンドポイントによって作動し得る文書としてウェブサービスコールに関するパラメータ及びデータをコード化してもよい。例えば、ｅＸｔｅｎｓｉｂｌｅＭａｒｋｕｐＬａｎｇｕａｇｅ（ＸＭＬ）又は他の適切なマークアップ言語のバージョンを、ウェブサービス要求文書をフォーマット化するために使用してもよい。一部の実施形態において、要求文書をフォーマット化するために使用されるマークアップ言語を、要求の処理を制御するパラメータの範囲を定めるが、他の実施形態において、マークアップ言語のある特徴は、それ自体（例えば、あるタグ）要求処理の態様を直接制御してもよい。さらに、一部の実施形態において、結果として得られる文書は、例えば、エンドポイントによるウェブサービス要求の処理を容易にするために、Ｓｉｍｐｌｅ Ｏｂｊｅｃｔ Ａｃｃｅｓｓ Ｐｒｏｔｏｃｏｌ（ＳＯＡＰ）のバージョンなどの他のプロトコル内にカプセル化されてもよい。

【００１５】

他のプロトコルが、さらにウェブサービスアーキテクチャの様々な実施形態内で用いられてもよい。例えば、ＷｅｂＳｅｒｖｉｃｅｓＤｅｓｃｒｉｐｔｉｏｎＬａｎｇｕａｇｅ（ＷＳＤＬ）のバージョンは、潜在的クライアントにインターフェースで接続する要求を公開するためにウェブサービスエンドポイントによって用いられる。ウェブサービスエンドポイントは、Ｕｎｉｖｅｒｓａｌ Ｄｅｓｃｒｉｐｔｉｏｎ，Ｄｉｓｃｏｖｅｒｙ ａｎｄ Ｉｎｔｅｇｒａｔｉｏｎ（ＵＤＤＩ）プロトコルのバージョンなどのディレクトリプロトコルを通じ、ウェブサービスを潜在的クライアントに知らせる場合がある。ウェブサービスインターフェースを介するコンピュータサービスの規定に関係する多くの他の種類のプロトコルが存在する場合があり、あらゆる所定のウェブサービス実装は当該プロトコルのあらゆる適切な組み合せを使用してもよい。

【００１６】

一部の実施形態において、インターフェース１０は、ウェブサービスインターフェースの代わり、又はウェブサービスインターフェースに加えて、ウェブサービスインターフェース以外のインターフェースをサポートしてもよいことに留意されたい。例えば、企業は、異なる種類のインターフェース（例えば、企業のイントラネットに対してカスタマイズされた、独自のインターフェース）を使用するであろう企業内のユーザのみならず、ウェブサービスプロトコルを介しサービスにアクセスするであろう企業外部のクライアントによって使用されるストレージサービスを実装してもよい。一部の実施形態において、インターフェース１０は、それを介してストレージサービスのあらゆるユーザがサービスにアクセスできるようなインターフェースで接続する様々な種類のそれぞれのプロトコルをサポートしてもよい。他の実施形態において、インターフェースの異なる例は、異なるインターフェースアプローチのために提供されてもよい。一部の実施形態において、クライアントとの情報のやり取り（例えば、サービス要求の受け取り及びサービス要求への応答）を取り扱うことに関係するインターフェース１０のこれらの態様は、ストレージサービス（例えば、バケット及びオブジェクトの階層へのサービスの体系）の一般的なアーキテクチャを実装するこれらの態様から独立して実施されてもよいことに留意されたい。このような一部の実施形態において、図２の説明とともに、さらに以下に詳しく説明するように、クライアントの情報のやり取り（例えば、ウェブサービスプロトコルを介した）のインターフェース１０の一部は、企業の内部などの特定のユーザによって迂回されてもよい。

【００１７】

図１に示すように、インターフェース１０は、ストレージサービスユーザにバケット２０へのアクセスを提供する。一般的に言えば、バケット２０はストレージサービスのユーザと関連するオブジェクト名前領域のルートとして機能してもよい。例えば、バケット２０はファイルシステムディレクトリ又はフォルダに類似していてもよい。一部の実施形態において、また個々のバケット２０は、ストレージサービスの使用量計算の基準となる場

合がある。例えば、ユーザは請求目的のために１つ以上のバケット２０と関連付けられる場合があり、ユーザは、これらのバケット２０によって設定された名前領域内に階層的に備わるストレージリソース（例えば、ストレージオブジェクト３０）の使用に対して請求されてもよい。

【００１８】

図で示した実施形態において、バケット２０ａ－ｎのそれぞれは、各アクセスポリシー２３ａ－ｎのみならず、関連するメタデータ２１ａ－ｎを含む。一般的に言えば、メタデータ２１が、所定のバケット２０の態様又は性質を表現するために使用される。例えば、メタデータ２１は、バケット作成日を識別する情報、その作成者の身元情報、バケットがそれに関連する任意のオブジェクト３０を有するか否かの情報、その他の適切な情報を含んでいてもよい。一部の実施形態において、メタデータ２１は、バケット２０に関連するオブジェクト３０の合計サイズ、バケット２０及び／又はその関連するオブジェクト３０に関するユーザのアクセス履歴、バケット２０に関連する請求履歴、又は、バケット２０の現行又はこれまでの使用に関係する他の適切な情報などの、バケット２０の使用特性を示す情報を含んでもよい。一実施形態において、各バケット２０は、ストレージサービスによって、ユーザによって、又は自動的に特定される、それぞれの固有の識別子と関連してもよい。固有の識別子は、メタデータ２１内、又はバケット２０の別個の性質、又はフィールドとして保存されてもよい。一部の実施形態において、所定のバケット２０は、明確な参照、ポインタ、又は他の所定のバケット２０と関連するオブジェクト３０と対応する情報を含まなくてもよい。むしろ、以下に更に詳しく説明するように、オブジェクト３０の位置及び選択は、キーマップとして本書で言及する別個のマッピング設備の使用を通じて実行されてもよい。

【００１９】

アクセスポリシー２３は、バケット２０に関連するオブジェクト３０へのアクセスを制御するために必要なあらゆる情報を含む。アクセスポリシー２３は、バケット２０及びその関連するオブジェクト３０へアクセスすることが許可された１人又は複数のクライアント、及びどのくらいの役割でアクセスするかを識別する情報を含んでもよい。例えば、アクセスポリシー２３は、１人以上のクライアントのために、ユーザ識別子及び／又は認証読み取りを許可されているかをさらに特定する。アクセスポリシー２３は、初期設定又はグループ指向規定（例えば、汎用読み取りアクセスを許可するがオブジェクト３０への書き込みアクセスを特定のクライアント又はクライアントグループに制限することによって）、又はあらゆる他の望ましいセキュリティモデルをさらに実行してもよい。

【００２０】

図で示した実施形態において、所定のバケット２０は、１つ以上のオブジェクト３０と関連され、各オブジェクトはそれぞれのメタデータ３１及びデータ３３を含んでもよい。一般的に言えば、オブジェクト３０のデータ３３はあらゆる一連のビットと対応してもよい。オブジェクト３０内に保存されるビットによって表されるデータの種類は、ストレージサービスに透過的であってもよい。すなわち、ビットはテキストデータ、実行可能プログラムコード、音声、ビデオ、又は画像データ、又はあらゆる他の種類のデジタルデータを表し、ストレージサービスは、オブジェクト３０を保存し操作することにおいて、これらの様々なデータの種類のうちから必ずしも識別することはない。一部の実施形態において、データ３３のサイズは固定上限（例えば１ギガバイト（ＧＢ））に限られてもよいが、他の実施形態において、オブジェクト３０は、ストレージサービスに使用可能な物理的ストレージリソースのみに従いサイズの拡大を許可してもよい。

【００２１】

バケット２１に関連するメタデータ２１と同様に、メタデータ３１は、対応するオブジェクト３０についてあらゆる望ましい記述的な情報を保存するように設定されてもよい。例えば、メタデータ３１は対応するオブジェクト３０が作成された日付及び／又は時間、オブジェクト３０のサイズ、オブジェクト３０によって保存されたデータ３３の種類（例えば、ＭｕｌｔｉｐｕｒｐｏｓｅＩｎｔｅｒｎｅｔＭａｉｌＥｘｔｅｎｓｉｏｎｓ（ＭＩ

ＭＥ）基準によって識別されるデータの種類）、又はあらゆる他の種類の記述的情報についての情報を含んでもよい。一部の実施形態において、メタデータ３１は、アクセスポリシー情報（例えば、オブジェクト３０に対して様々なユーザが有してもよいアクセスの種類を示す許可情報）、オブェクト費用情報（例えば、オブジェクト３０と関連する請求割合又は履歴）、又は、オブジェクト３０に起因するあらゆる他の適切な情報又は情報の種類の組み合せのみならず、オブジェクト３０と対応するユーザ対話をしめす使用情報又は履歴情報を保存してもよい。一部の実施形態において、クライアントはメタデータ３１として保存されるオブジェクトデータと共にメタデータを提供してもよいが、他の事例においては、メタデータ３１は、ストレージサービス特徴（例えば、図２に示され、以下に説明されるストレージサービスシステム）を管理するシステムによって作成されるメタデータを含んでもよい。メタデータ３１の一部、全部、又はいずれでもないものが、メタデータの種類、クライアントのアクセス権の特定の規定、又は他の適切な要因にしたがって、オブジェクト３０へのアクセス権を有するクライアントにアクセス可能であってもよい。

【００２２】

一実施形態において、個々のオブジェクト３０は、情報の２つの相異なるアイテム、キー又はロケータの、いずれかを使用するストレージサービスシステム内で識別されてもよい。一般的に言えば、キー及びロケータは異なる手段で解釈してもよいが、キー及びロケータは、全体としてのストレージサービスシステムの名前領域の内容内で解釈されてもよい英数字文字列又は他の種類の記号をそれぞれに含んでもよい。一実施形態において、キーは、特定のバケット２０（例えば、新規オブジェクトを保存するためのクライアントからの要求に応じて）内に対応するオブジェクト３０が作成された時にクライアントによって特定されてもよい。ユーザによってキーが特定されない場合、キーは、ストレージサービスシステムによって新規オブジェクト３０に割り当てられる。そのような実施形態において、特定のバケット２０のオブジェクト３０と関連する各それぞれのキーは、そのバケット２０の名前領域内固有であることが求められる場合がある。一般的に言うと、キーは、対応するオブジェクトがストレージサービスシステム内に存在する限り、クライアントが、対応するオブジェクト３０にアクセスする際に介する有効な識別子として存続してもよい。

【００２３】

所定のバケット２０内で、キーは、従来のオペレーティングシステムのファイルシステムに共通のファイルディレクトリ、又はフォルダ名前領域と類似の階層的オブジェクト名前領域を作成するために使用されてもよい。例えば、クライアントは、固有の識別子０５０７３９５１７を有する特定のバケット２０へのアクセス権を読み込む、又は書き込むオブジェクトを与えられてもよい。一実施形態において、クライアントは、バケット内でオブジェクトと対応するバケット名前領域内のキーを作成するために、ｈｔｔｐ：／／ｓｔｏｒａｇｅｓｅｒｖｉｃｅ．ｄｏｍａｉｎ．ｃｏｍ／０５０７３９５１７へウェブサービスコールを発行してもよい。例えば、クライアントは、「Ｍｙ　Ｄｏｃｕｍｅｎｔｓ／Ｅｍａｉｌ／ｍｅｓｓａｇｅ．ｔｘｔ」というキーを使用するこの特定のバケット内でオブジェクト３０が作成されることを特定してもよく、このようなオブジェクト３０は、アドレス

ｈｔｔｐ：／／ｓｔｏｒａｇｅｓｅｒｖｉｃｅ．ｄｏｍａｉｎ．ｃｏｍ／０５０７３９５１７／ＭｙＤｏｃｕｍｅｎｔｓ／Ｅｍａｉｌ／ｍｅｓｓａｇｅ．ｔｘｔ

へのウェブサービスコールを使用してアクセスされてもよい。

【００２４】

一部の実施形態において、キーによって暗示される階層的構造は、オブジェクトストレージの基礎をなす階層に必ずしも反映されなくてもよいことに留意されたい。例えば、一実施形態において、所定のバケット２０と関連するオブジェクト３０は、オブジェクト３０と関連するキーが階層を暗示しているにもかかわらず、ストレージサービスシステム内で平滑な、非階層な形で保存されてもよい。すなわち、このような実施形態において、バケット２０は、他のバケット２０を階層的に含まなくてもよい。しかしながら、他の実施

形態において、他のバケット２０内のバケット２０の階層的包含は、オブジェクトキーによって暗示される階層に対して直接一致する必要はないが、バケットのいかなる当該階層がサポートされてもよい。

【００２５】

一実施形態において、キーによって識別されたオブジェクト３０へアクセスするためのクライアントによる要求は、要求されたオブジェクト３０の基礎をなすデータ３３が取り出される又は修正される前に、クライアント認証手順、アクセス制御チェック、及び／又はマッピング過程（以下にさらに詳しく説明するような）を受ける場合がある。例えば、クライアントは、クライアントの身元を証明するためにパスワード又は他の資格を提供することを要求される場合があり、一度識別されると、要求されるバケット２０と関連するアクセス制御パラメータは、識別されたクライアントが要求されたキーへのアクセスを保証する十分な権限が与えられているかどうかを判断するために評価されてもよい。それに反して、ストレージサービスシステムは、キーよりもむしろロケータによってオブジェクト３０にアクセスする代替の方法をサポートしてもよい。一般的に言うと、ロケータは、ストレージサービスシステムに既知のすべてのオブジェクト３０の中から、オブジェクト３０の世界的に固有の識別子を表してもよい。すなわち、キーが、特定のバケット２０と関連する名前領域に対して固有であってもよいが、ロケータはすべてのバケット２０内のすべてのオブジェクト３０の世界的名前領域内で固有であってもよい。例えば、ロケータは、他のロケータの中で固有であるべきストレージサービスシステムによって作成される英数文字列を含んでもよい。さらに以下に詳しく説明するように、一部の実施形態において、オブジェクト３０の複数インスタンスは、例えば、データ冗長性及びフォルトトレランスを増加するために、ストレージサービスシステムを実装するために使用された物理的ストレージデバイスの全体を通じて複製されてもよい。そのような実施形態において、固有のロケータは、所定のオブジェクト３０の各複製されたインスタンスに対して存在してもよい。

【００２６】

一部の実施形態において、キーは、オブジェクト３０がストレージサービスシステム内に存在する限り、オブジェクト３０へのアクセスに有効でありつづけることが保証されてもよく、当該保証はそのオブジェクト３０のあらゆる所定のロケータに適応されてもよく、又は適応されなくてもよい。例えば、オブジェクト３０の複製されたインスタンス（又は複製）が、異なる物理的なストレージの位置（例えば、その基礎となるストレージ媒体の障害又は置換によって）に移動する場合、その新規の位置におけるオブジェクト３０の移動したインスタンスと対応する他のロケータが作成され、使用されることがあったとしても、特定のインスタンスを参照するロケータは有効ではなくなってもよい。キー及びロケータ間の関係における更なる詳細は、キーマップシステムコンポーネントの操作についての論考で提供される。

【００２７】

キー系対ロケータ系オブジェクトアクセスの例として、オブジェクト３０は、上記キーによって参照され、ｈｔｔｐ：／／ｓｔｏｒａｇｅｓｅｒｖｉｃｅ．ｄｏｍａｉｎ．ｃｏｍ／０５０７３９５１７／Ｍｙ　Ｄｏｃｕｍｅｎｔｓ／Ｅｍａｉｌ／ｍｅｓｓａｇｅ．ｔｘｔは、ストレージサービスシステム内に保存される１つ以上のインスタンスを有することがあり、その１つは、ｈｔｔｐ：／／ｓｔｏｒａｇｅｓｅｒｖｉｃｅ．ｄｏｍａｉｎ．ｃｏｍ／ｌｏｃａｔｏｒ／３８５９Ｃ８９Ａ２０８ＦＤＢ５Ａという形式のロケータによって識別されてもよい。

この特定の実施形態において、オブジェクト３０に対するキー参照は特定のバケット２０に関連して表現されるが、ロケータ参照は、世界的ロケータ空間（他の種類のロケータコード化又は形式が用いられてもよいが）内の１２８ビットの１６進絶対数として表現されることに留意されたい。一実施形態において、ロケータに命令されたクライアント発行のウェブサービス要求は、すべての認証、アクセス権、翻訳、又はキー系のウェブサービス要求に適応されてもよい他のステップのいくつか、又はすべてを回避してもよい。処理の

より少ない層のため、一部のかかる実施形態において、ロケータ系の要求はキー系の要求よりも更に早く処理されてもよい。しかしながら、セキュリティ対策は、ロケータ系要求のために回避されることがあり、クライアントは、敏感なオブジェクト３０が改ざんされていないことの個人的保証を（例えば、ロケータの送信及び受信において暗号化された、又は他の方法を使用し）提供する必要がある場合がある。さらに、ロケータの持続は保証されないため（例えば、上記記載のオブジェクトインスタンスの移動の場合）、ロケータ系オブジェクトアクセスの操作を選択するクライアントは、例えば、プリエンプティブ系において、又は存在するロケータがもはや有効ではないことを発見することに応じて新規ロケータを取得することによって、ロケータが使用中に無効になる可能性を許容することが必要となってもよい。

【００２８】

クライアントのストレージの必要性及び上記の注意点によって、ロケータ系アクセスは、キー系アクセスに関連してより一層の処理能力（例えば、ウェブサービス要求処理の待ち時間及び処理量において）を提供してもよい。例えば、クライアントは、特に敏感でない頻繁にアクセスされるオブジェクト３０を参照するためにロケータ系アクセスを使用することを選択してもよい。一部の実施形態において、ロケータ系アクセスは個々のオブジェクト３０に基づき不可能であることがあり、したがって、当該のオブジェクトにアクセスすることを望むクライアントに、キー系要求を使用し、当該の要求と関連するあらゆる認証及びアクセス権制御を提出するように強制することに留意されたい。しかしながら、ロケータ系アクセスが可能になったオブジェクト３０に対しても、有効なロケータを保持しない悪意のある、又は機能不良のクライアントは、あらゆる所定のオブジェクト３０にうまくアクセスする任意の可能性だけを有してもよい。このような可能性は、大規模ロケータ名前領域、ロケータを作成する安全技術（例えば、オブジェクトデータの安全ハッシュの使用）、又は他の適切な技術の使用を通じて任意に防ぐことができる。

【００２９】

ストレージシステムアーキテクチャ及び実装

図１に示すようなウェブサービス系のストレージサービスを実装するように設定されてもよい、ストレージサービスシステムアーキテクチャの一実施形態を図２に示す。図で示した実施形態において、多くのストレージクライアント５０ａ－ｎは、ネットワーク６０を介してウェブサービスプラットフォーム１００と情報をやり取りするように設定されてもよい。ウェブサービスプラットフォーム１００は、ストレージサービスコーディネータ１２０（又は単に、コーディネータ１２０）の１つ以上のインスタンスとインターフェースで接続するように設定されてもよく、ストレージサービスコーディネータは、１つ以上のキーマップインスタンス１４０及びビットストアノード１６０とインターフェースで入れ替わりに接続してもよい。さらに、レプリケータ１８０は、レプリケータキーマップインスタンス１９０のみならず、ビットストアノード１６０とインターフェースで接続するように設定されてもよい。コーディネータ１２０とレプリケータ１８０の両方は、ノードピッカー１３０とインターフェースで接続してもよい。図で示した実施形態において、ノードピッカー１３０、キーマップ１４０、ビットストアノード１６０及びレプリケータキーマップ１９０の各インスタンスは、発見及び障害検出デーモン（ＤＦＤＤ）１１０のそれぞれのインスタンスと関連してもよい。所定のコンポーネントの１つ以上のインスタンスが存在する場合には、以下で言及するコンポーネントは単数又は複数のいずれかであってもよい。しかしながら、いずれの形式の使用も他方を不可能にすることを意図しない。

【００３０】

様々な実施形態において、図２に示すコンポーネントは、コンピュータハードウェア（例えば、マイクロプロセッサ、又はコンピュータシステム）、又はこれらの技術の組み合せによって指示が直接、又は非直接実行可能である場合、コンピュータハードウェア内で直接実装されてもよい。例えば、図２のコンポーネントは、図２９に示すコンピュータシステムの実施形態、及び以下に説明するようなコンピュータ関係のノード（又は単にノード）の多くを含む分散型システムによって実装されてもよい。さまざまな実施形態におい

て、所定のストレージサービスシステムコンポーネントの機能性は、特定のノード又はいくつかのノードに渡って分散されることによって実装されてもよい。一部の実施形態において、所定のノードは、１つ以上のストレージサービスシステムコンポーネントの機能性を実装してもよい。図２のコンポーネントの一般的な機能性の概略及び図３に示すようなストレージサービスシステムの典型的な物理的配置に続き、特定のストレージシステムコンポーネントの一部の実施形態を図４から図２８に説明とともに以下に提供する。

【００３１】

一般的に言えば、ストレージクライアント５０は、ネットワーク６０を介するウェブサービスプラットフォーム１００にウェブサービス要求を提出するように構成されるクライアントのあらゆる種類を網羅する。例えば、所定のストレージクライアント５０は、ウェブブラウザの適切なバージョン、又はプラグインモジュール、又はウェブブラウザによって提供される実行可能環境に対する、又は環境内の拡張子を実行するように構成される、他の種類のコードモジュールを含んでもよい。あるいは、ストレージクライアント５０は、データベースアプリケーション、メディアアプリケーション、オフィスアプリケーション又は持続的ストレージリソースを使用するあらゆる他のアプリケーションなどのアプリケーションを網羅してもよい。一部の実施形態において、当該アプリケーションは、すべての種類のウェブ系データに対する完全なブラウザサポートを必ずしも実装することのない、ウェブサービス要求を作成し、処理するための十分なプロトコルサポート（例えば、ＨｙｐｅｒｔｅｘｔＴｒａｎｓｆｅｒＰｒｏｔｏｃｏｌ（ＨＴＴＰ）の適切なバージョンに対する）を含んでもよい。すなわち、ストレージクライアント５０は、ウェブサービスプラットフォーム１００と直接対話するように構成されるアプリケーションでもよい。以下に説明するように、ストレージクライアント５０は、ＲｅｐｒｅｓｅｎｔａｔｉｏｎａｌＳｔａｔｅＴｒａｎｓｆｅｒ（ＲＥＳＴ）形式ウェブサービスアーキテクチャ、文書系、又はメッセージ系ウェブサービスアーキテクチャ、又は他の適切なウェブサービスアーキテクチャに従いウェブサービス要求を作成するように設定されてもよい。

【００３２】

他の実施形態において、ストレージクライアント５０は、これらのアプリケーションに透過的である方法における他のアプリケーションに対しウェブサービス系ストレージへのアクセスを提供するように設定されてもよい。例えば、ストレージクライアント５０は、上記に記載されるストレージモデルの適切な改良型に従い、ストレージを提供するためにオペレーティングシステム又はファイルシステムを統合するように設定されてもよい。しかしながら、オペレーティングシステム、又はファイルシステムは、ファイル、ディレクトリ、及び／又はフォルダの従来型ファイルシステム階層などの、アプリケーションに対する異なるストレージインターフェースを提示してもよい。このような実施形態において、アプリケーションは、図１のストレージシステムサービスモデルを使用するように修正される必要がなくてもよい。代わりに、ウェブサービスプラットフォーム１００へのインターフェース接続の詳細は、オペレーティングシステム環境内で実行されるアプリケーションに代行して、ストレージクライアント５０及びオペレーティングシステム、又はファイルシステムによって統合されてもよい。

【００３３】

ストレージクライアント５０は、ネットワーク６０を介するウェブサービスプラットフォーム１００へのウェブサービス要求を送り、応答を受信してもよい。様々な実施形態において、ネットワーク６０は、クライアント５０とプラットフォーム１００間のウェブ系の通信を確立するために必要なネットワーキングハードウェア及びプロトコルのあらゆる適切な組み合せを網羅してもよい。例えば、ネットワーク６０は、インターネットを集合的に実装する様々な電気通信網及びサービスプロバイダを概して網羅する。ネットワーク６０は、公共又は私的ワイアレスネットワークのみならず、ローカルエリアネットワーク（ＬＡＮ）又は広域エリアネットワーク（ＷＡＮ）などの私的ネットワークを含んでもよい。例えば、所定のクライアント５０及びウェブサービスプラットフォーム１００の両方は、独自の内部ネットワークを有する企業内においてそれぞれ支給されてもよい。そのよ

うな実施形態において、ネットワーク６０は、インターネットとウェブサービスプラットフォーム１００間のみならず、所定のクライアント５０とインターネット間におけるネットワーキングリンクを確立するために必要な、ハードウェア（例えば、モデム、ルータ、スイッチ、負荷分散装置、プロキシ、サーバなど）及びソフトウェア（例えば、プロトコルスタック、会計ソフトウェア、ファイアーウォール／セキュリティソフトウェアなど）を含んでもよい。一部の実施形態において、ストレージクライアント５０は、公的インターネットよりはむしろ私的ネットワークを使用してウェブサービスプラットフォーム１００と通信してもよいことに留意されたい。例えば、クライアント５０は、ストレージサービスシステムとして同一の企業内で支給されてもよい。そのような場合、クライアント５０は、私的ネットワーク６０の全体を通じてプラットフォーム１００に通信してもよい。このような場合、クライアント５０は、私的ネットワーク６０全体を通じてプラットフォーム１００と通信してもよい（例えば、公的にアクセス可能でない、インターネットベースのコミュニケーションプロトコルを使用してもよいＬＡＮ、又はＷＡＮ）。

【００３４】

一般的に言うと、ウェブサービスプラットフォーム１００は、ストレージサービスシステムによって保存されるオブジェクト３０にアクセスするための要求のような、ウェブサービス要求を受信し、処理するように構成される、１つ以上のエンドポイントを実装するように設定されてもよい。例えば、ウェブサービスプラットフォーム１００は、前述の例において使用されたエンドポイントであるｈｔｔｐ：／／ｓｔｏｒａｇｅｓｅｒｖｉｃｅ．ｄｏｍａｉｎ．ｃｏｍを実装するように構成されるハードウェア及び／又はソフトウェアを含んでもよく、このようなエンドポイントに命令されたＨＴＴＰ系ウェブサービス要求は適切に受信され処理される。一実施形態において、ウェブサービスプラットフォーム１００は、クライアント５０からウェブサービス要求を受信し、それをコーディネータ１２０、又は処理のためのストレージサービスシステムの他のコンポーネントへ転送するように構成されるサーバシステムとして実装されてもよい。他の実施形態において、ウェブサービスプラットフォーム１００は、大規模ウェブサービス要求プロセス負荷を動的に管理するように構成される、負荷バランシング及び他の要求管理特徴を実装する多くの別個のシステム（例えば、クラスタトポロジーにおいて）として設定されてもよい。

【００３５】

様々な実施形態において、ウェブサービスプラットフォーム１００は、上記に詳述されるように、ＲＥＳＴ形式、又は、文書系（例えば、ＳＯＡＰ系）の種類のウェブサービス要求をサポートするように設定されてもよい。１つの特定の実施形態において、プラットフォーム１００は、ストレージサービスシステムによって管理されるエンティティの様々な操作をサポートする、特定のウェブサービスアプリケーションプログラミングインターフェース（ＡＰＩ）を実装するように設定されてもよい。例えば、プラットフォーム１００によって実装されるＡＰＩは、バケット２０又はオブジェクト３０のリスト（フィルタパターン又は基準に従い任意にフィルタされる）、バケット２０又はオブジェクト３０のデータ又はメタデータの取り出し、及びバケット２０又はオブジェクト３０の作成又は削除を含む、バケット又はオブジェクト上の基本的なクライアント操作をサポートしてもよい。一部の実施形態において、ＡＰＩは、複数のバケット２０又はオブジェクト３０に対する操作のバッチアプリケーションなどの、さらに洗練されたクライアント操作をサポートしてもよい。

【００３６】

クライアントのウェブサービス要求のためのアドレス可能なエンドポイントとして機能することに加え、一部の実施形態において、ウェブサービスプラットフォーム１００は、様々なクライアント管理特徴を実装してもよい。例えば、プラットフォーム１００は、要求するクライアント５０の身元を追跡し、クライアント要求の数及び／又は頻度、クライアント５０に代行して保存された、又は取り出されたオブジェクト３０のサイズ、クライアント５０によって使用された全体的なストレージ回線容量、クライアント５０によって要求されたストレージのクラス、又はあらゆる他の測定可能なクライアント使用パラメー

タなどの、ストレージリソースを含む、ウェブサービスのクライアント使用の計測及び会計を統合してもよい。プラットフォーム１００は、財務会計及び請求システムをさらに実装、又はクライアント使用活動の報告及び請求のための外部システムによって要求又は処理される使用データのデータ系を維持してもよい。

【００３７】

一部の実施形態において、プラットフォーム１００は、クライアント５０から受信した要求の割合及び種類を反映する評価指数、当該要求によって使用された回線容量、当該要求に対する遅延処理手続きシステム、システムコンポーネント使用（例えば、ストレージサービスシステム内のネットワーク回線容量及び／又はストレージ使用）、要求の結果によりエラーの割合及び種類、要求されたオブジェクト３０の特徴（例えば、サイズ、データの種類など）、又はあらゆる他の適切な評価指標をなどの様々なストレージサービスシステム操作評価指標を収集及び／又は監視するように設定されてもよい。このような実施形態において、プラットフォーム１００は、例えば、平均超過時間として、又は様々な分析に従ってもよい特定のデータポイントとして、当該評価指数を総計に収集するように設定されてもよい。様々な実施形態において、当該の評価指数は、クライアント５０に可視である、又は不可視であってもよい方法でシステムの能力を試験又は監視するために用いられてもよい。例えば、一実施形態において、当該評価指数は、システムコンポーネントを調整、及び維持するためのシステム管理者によって使用されてもよいが、他の実施形態において、当該評価指数（又は当該評価指数と同等の一部）は、ストレージサービスシステムの使用を当該クライアントが監視できるようにクライアント５０に使用可能にされてもよい。

【００３８】

一部の実施形態において、プラットフォーム１００は、ユーザ認証及びアクセス制御手順をさらに実装してもよい。例えば、所定のバケット２０と関連する特定のオブジェクト３０へアクセスするための所定のウェブサービス要求について、プラットフォーム１００は、要求に関連するクライアント５０が、所定のバケット２０及び特定のオブジェクト３０へのアクセスする権限が与えられているかどうかを確かめるように設定されてもよい。プラットフォーム１００は、例えば、身元、パスワード、又は所定のバケット２０に関連する資格者に対する他の資格の評価すること、特定のオブジェクト３０への許可できる操作を指定する、アクセス制御リストに対する特定のオブジェクト３０に対し要求されたアクセスを評価することにより、当該権限を判断してもよい。クライアント５０が、バケット２０へのアクセスをする、又はオブジェクト３０に要求された操作（例えば、クライアント５０が、読み込みアクセス特権のみを有するオブジェクト３０の書き込みを試みる）を行う十分な資格を有しない場合、プラットフォーム１００は、例えば、要求するクライアント５０にエラー状態を示す応答を返すことにより、対応するウェブサービス要求を拒否してもよい。一部の実施形態において、各バケット２０及びオブジェクト３０は、そのバケット又はオブジェクトにアクセスすることを統制する、関連するアクセス制御規定を有してもよいことを考慮されたい。当該アクセス制御規定は、メタデータ２１又は３１内にアクセス制御情報の記録又はリストとして、あるいはメタデータ２１及び３１から別個のデータ構造として保存されてもよい。

【００３９】

一部の実施形態において、図２のシステムのようなストレージサービスシステムは、任意サイズのオブジェクト３０をサポートしてもよいが、他の実施形態において、オブジェクト３０は、チャンクサイズ(chunk size)ともいわれる、ある最大サイズを強制されることもある。このような一部の実施形態において、クライアントが、キーと関連して保存すべきデータを提供し、そのデータがチャンクサイズを上回る場合、プラットフォーム１００は、チャンクサイズに従い、データを２つ以上のチャンクに分配するように設定されてもよい。一実施形態において、プラットフォーム１００は、関連するキー値を有するそれぞれのオブジェクト３０として各チャンクを作成するように設定されてもよい。プラットフォーム１００は、クライアント提供キーを参照するアクセスのための要求が行われた場

合、元のクライアントデータが、チャンクから再構築されることが出来るような方法で、クライアント提供キーの機能として各チャンクに対するキー値を作成してもよい。例えば、プラットフォーム１００は、クライアントデータからＮチャンクを作成するように構成され、クライアント提供キーに対し、Ｎの相異なるパターンを付加することによってこれらのチャンクのためのＮ対応キーを作成し、Ｎの相異なるパターンは、Ｎチャンクが作成されたのと同一の順序で辞書編集的に順序付けられてもよい。Ｎチャンクのそれぞれは、以下に説明する技術を使用する相異なるオブジェクト３０として管理され、元のデータは、クライアント提供キーがプレフィックスであるようなキー値を有するすべてのオブジェクト３０をリストすることによって、またこれらのオブジェクトをリストされた順序で取り出すことによって作成されてもよい。一部の実施形態において、個々のチャンクは、他のチャンクを阻害することなくアクセスされ、修正され、又は取り除かれることがあり、１つの大きいオブジェクト３０としてのデータを維持することに関係してシステムの能力を改善してもよい。一部の実施形態において、クライアント５０は、それが提供するデータオブジェクトをチャンクに分割すべきかどうかを指定することを許可されてもよい。

【００４０】

図２に示すストレージサービスシステムコンポーネントの多くと同様に、他のコンポーネントからのウェブサービスプラットフォーム１００の機能性を分離することは、ストレージサービスシステムの保守管理及び全体的な拡張性を改善してもよい。例えば、付加的なハードウェア及びソフトウェア供給源は、他のタスクに割り振られた供給源の付加的なウェブサービス処理負荷を独立して維持する為に特に提供されてもよい。さらに、プラットフォーム１００と関連するあらゆる供給源の障害の影響は、その特定の機能領域の範囲内にとどめられる場合があり、したがって、障害の隔離と解決を容易にする。しかしながら、一部の実施形態において、プラットフォーム１００の機能性は、他のコンポーネントにおいて統合されてもよいことを考慮されたい。例えば、コーディネータ１２０は、プラットフォーム１００と関連するタスクを含むように設定されてもよい。

【００４１】

ウェブサービスプラットフォーム１００は、クライアント５０がストレージサービスシステムの特徴にアクセスすることを通じ、プライマリインターフェースを示すが、当該特徴にたいする唯一のインターフェースを示す必要がないことに留意されたい。例えば、一部の実施形態において、コーディネータ１２０は、ウェブサービスインターフェースから相異してもよい代替ＡＰＩをサポートするように設定されてもよい。このような代替ＡＰＩは、例えば、企業の内部のクライアントが迂回ウェブサービスプラットフォーム１００に対するストレージシステムを提供することを可能にするために使用されてもよい。ある事例において、プラットフォーム１００の会計及び／又は信任は、管理クライアントなどの内部クライアントに必要がなくてもよい。

【００４２】

コーディネータ１２０は、ウェブサービスプラットフォーム１００及びストレージサービスシステムの他のコンポーネントとの間の行動を協調させるように設定されてもよい。一実施形態において、コーディネータ１２０の主要な義務は、これらのオブジェクト３０に命令されたウェブサービス要求に応じ、オブェクト３０に対するオブェクトデータ３３及びメタデータ３１の読み込み及び書き込み動作の指示を含んでもよい。例えば、以下に詳しく説明するように、オブェクト読み込みアクセスは、所定のオブジェクト３０の複製が保存される、ビットストアノード１６０を示すロケータを取り出すために、キーマップインスタンス１４０へのアクセスを行うことに続き、要求されたデータを読み込むために特定のビットストアノード１６０へアクセスすることを行うことを必要としてもよい。同様に、オブジェクト作成及び修正は、必要であれば、作成又は修正されたロケータを反映するために、様々なビットストアノード１６０へオブジェクト３０の多数の複製を保存し、キーマップインスタンス１４０を更新することを必要としてもよい。一部の実施形態において、コーディネータ１２０は、キーマップインスタンス１４０及びビットストアノード１６０へのこれらの読み込み及び書き込み操作を行うように構成されてもよい。しかし

ながら、一部の実施形態において、コーディネータ１２０は、作成又は修正時点においてオブジェクト３０の所望の複製の全数を作成することを行わなくてもよいことに留意されたい。さらに以下に詳しく説明するように、一部の実施形態において、オブジェクト３０に対する書き込み操作は、コーディネータ１２０が、そのオブジェクト３０の複製のある数の書き込み（たとえば、２つの複製）を完了した時に終了したと見なされてもよい。さらに、このオブジェクト３０の複製は、レプリケータ１８０による帯域外又は非同期操作によって完了されてもよい。すなわち、このような実施形態において、オブジェクト作成又は修正操作の帯域内又は同期部分は影響されたオブェクト３０の複製の合計所望数より少ない作成物を含んでもよい。コーディネータ１２０は、キーマップインスタンス１４０、ビットストアノード１６０、及び他のシステムコンポーネントから相異なるコンポーネントとして示されるが、一部の実施形態において、コーディネータ１２０のインスタンスについて、他のシステムコンポーネント（例えば、単独のコンピュータシステムによって実行されるソフトウェアコンポーネントとして）と共に実装されることは可能であるという点に留意されたい。したがって、本書の定義は、コーディネータ１２０がビットストアノード１６０、キーマップインスタンス１４０、又は別のコンポーネントへデータを保存し、又はそこからデータを取り出すことに言及してもよいが、一部の実施形態において、このような処理は共有コンピューティングシステム供給源内で起こってもよいことを理解されたい。

【００４３】

図１に関し、上記に記載されるとおり、一部の実施形態において、ストレージサービスシステムは、バケット系のストレージモデルを含むことがあり、様々なオブジェクト３０のためのキーが、管理（例えば、会計、請求）、セキュリティ、又はその他の目的のためにバケット２０内にグループ分けされてもよい。一実施形態において、コーディネータ１２０は、クライアント５０からのウェブサービス要求に応じ、様々なバケット関連の操作を処理するように設定されてもよい。例えば、コーディネータ１２０は、以下のバケット操作のいくつか、又はすべてを行うように設定されてもよい。

－バケットの作成：バケット２０のための新しいバケットネームを作成及び保存する。

－非空バケットの削除：関連するメタデータ２１を含む所定のバケット２０、及び所定のバケット２０内のオブジェクト３０と関連するすべてのキーを削除する。

－空バケットの削除：所定のバケット２０と関連するオブェクト３０のキーがない場合のみ、所定のバケット２０及び関連するメタデータ２１を削除し、それ以外はエラー状態を返す。

－バケットの書き込み：存在するバケット２０にデータを書き込む（例えば、メタデータ２１）。

－バケットキーのリストアップ：所定のバケット２０（パターン、正規表現、ワイルドカードなどに従い任意に保存され、又はフィルタされた）と関連するオブジェクト３０のキーをリストする。

－バケットのリストアップ：所定の加入者（例えば、ユーザ又はクライアント５０）と関連するバケットをリストする。

一部の実施形態において、コーディネータ１２０は、コリジョン作成の低い確率を有する適切な乱数アルゴリズムを使用し、新しく作成されたバケット２０に対する識別子を作成するように設定されてもよい。他の実施形態において、コーディネータ１２０は、例えば、バケット作成に対するクライアント要求における存在するバケット識別子に関し、固有性に対する要求された識別子を調べることによって、クライアント特定バケット識別子をサポートするように設定されてもよい。

【００４４】

上記に記載されるように、オブジェクト３０のインスタンスは、例えば、オブジェクトデータが、あらゆる所定のノード１６０又はそれと関連するインフラストラクチャの障害をしのぐ可能性を増加させるために、所定のノード１６０又は異なるビットストアノード１６０に複製されてもよい。ストレージサービスシステム内のオブジェクトの複製は、以

下のように、ノードピッカー１３０及びレプリケータ１８０による図示した実施形態で扱われてもよい管理及び最適化のためのいくつもの機会を呈する。

【００４５】

コーディネータ１２０がオブジェクト３０を書き込む要求を受け取る時、コーディネータは、書き込みが完了したことを表す前にノード１６０の所定の数にオブジェクト３０を相応して書き込んでもよい。しかしながら、書き込まれるべきオブジェクト３０に対するノード１６０の数と特定の選択は、異なるストレージ規定検討の数によって異なる場合がある。例えば、書き込み操作が完了したことが見なされる前に、オブジェクト３０の複製のある最低数（例えば、２つ又は３つ）がうまく書き込まれることを要求することは、可能性のある障害を考慮すると、書き込みデータが永続的であるためには賢明である場合がある。しかしながら、複製の最低数を保存するために選ばれたノード１６０が、障害の異なる可能な場所又は領域から分散されることを確実にすることも望ましい場合がある。例えば、同一のデータセンタに位置付けられるノード１６０は、地理的に隔てられたノード１６０よりも同時に（例えば、自然災害、電力障害などの壊滅的な障害によって）機能しなくなる可能性が高い。

【００４６】

一般的にストレージノード選択論理といわれることもあるノードピッカー１３０は、コーディネータ１２０及びレプリケータ１８０によってアクセス可能なサービスとして構成されることがあり、一実施形態において、様々なストレージ規定を満たすようなオブジェクト読み込み及び書き込み操作のためにノード１６０の選択のためのアルゴリズムを実装してもよい。例えば、上記に概説したオブジェクト３０の書き込みの場合において、ノードピッカー１３０は書き込みプラン、又はオブジェクト３０を書き込むべきノード１６０の特定の配列を構築するために操作してもよい。特定の書き込みプランを立てることにおいて、ノードピッカー１３０は、書き込みプランが成功の妥当な可能性を有することを確実にするように設定されてもよく、例えば、書き込みプランで特定されたノード１６０は、実際に操作可能であり、オブジェクト３０を受け入れるために使用可能である十分なストレージリソースを有することが期待され、完了した場合、その書き込みプランは、書き込み操作に関連するすべてのストレージ規定を満たす。書き込みストレージ規定の例は以下を含む。

－永続性規定：書き込みプランがうまく完了した場合、オブジェクト３０のインスタンスは、少なくともＮ相違ノード１６０に保存される。

－領域多様性規定：可能であれば、書き込みプランは、少なくともＭ相違領域間に分散されるノード１６０を含む。

－ローカル性規定：可能であれば、書き込みプランは、要求するコーディネータ１２０にたいしてローカルである領域のノード１６０に、選択（例えば、数で）を与える。

－負荷平衡規定：ノード１６０間の書き込み要求トラフィックを均一にするよう試みる（例えば、「ホットノード」を避けるために）。

－空間平衡規定：ノード１６０間の保存供給源容量の使用を均一にするよう試みる。

－最低コスト連鎖規定：書き込みプランにおけるノード書き込み操作の配列の合計コスト（例えば、ネットワーク遅延）を最小限にするよう試みる。

【００４７】

様々な実施形態において、ノードピッカー１３０は、所定の書き込みプランを立てる時、これらの規定のいくつか又はすべて、又はリストアップされない他の規定を考慮に入れることに留意されたい。さらに、異なる規定が、異なる優先事項によって重要性を増す場合がある。例えば、一実施形態において、持続性規定は、他の規定が最善努力系で満たされることがある一方、すべての書き込みプランが満たされなければならない義務的規定であってもよい。一部の事例において、一部のストレージ規定は他と拮抗してもよい。例えば、異なる領域間にオブジェクトインスタンスの幅広い分散を助ける領域多様性規定は、概して特定の領域内にオブジェクトインスタンスをローカル化するローカル性規定に反する。オブジェクトインスタンスの数が十分に大きい場合、両規定を満たすことが可能であ

る場合がある。例えば、オブジェクト３０の５つのインスタンスが作成された場合、要求するコーディネータ１２０にローカル的である２つのインスタンスを２つの相異なる領域に保存し、３つのインスタンスを３番目の相異なる領域に保存し、したがって、ローカル性規定及び領域多様性規定の両方を満たすことが可能である場合がある。書き込みプランを特定するすべての規定を満たすことが不可能でなければ、ノードピッカー１３０は、満たされるこれらの規定を優先させるよう試み、最善書き込みプランを作成し、又はオブジェクト書き込みが十分に行われないことを示す要求するコーディネータ１２０に対するエラー表示を返してもよい。

【００４８】

また一部の実施形態において、ノードピッカー１３０は、読み込みオブジェクト３０のコーディネータ１２０を援助してもよい。例えば、オブジェクト読み込み操作は、もともと、又は最も最近に書いた要求されたオブジェクト３０で一部のコーディネータ以外のコーディネータ１２０によって要求されてもよい。したがって、書き込んでいるコーディネータ１２０に関連しローカルで保存されてもよいオブジェクト３０のインスタンスは、読み込んでいるコーディネータ１２０に関連してローカルではなくてもよい。ノードピッカー１３０は、読み込んでいるコーディネータ１２０に使用可能の読み込み操作を提供し得るノード１６０を識別するように設定されてもよい。例えば、ノードピッカー１３０は、読み込んでいるコーディネータ１２０に最も近い（例えば、地理的距離又はネットワークトポロジーに関して）ノード１６０、又は最も高い読み込み回線容量を提供するノード１６０（例えば最小負荷ノード１６０又はストレージハードウェアのより高い能力クラスを有するノード１６０）を識別してもよく、又はノードピッカー１３０は、オブジェクト３０を読み込むノード１６０を選択するための他の性能基準を使用してもよい。他の実施形態において、コーディネータ１２０の読み込みに関する読み込みの操作を最適化するよりは、ノードピッカー１３０は、全体としての（例えば、広域読み込みスループットを最大限にするために）性能を最適化するために、同時読み込み操作を広域的にプランしてもよい。

【００４９】

書き込みプランを立て、オブジェクト読み込み操作に関連してコーディネータ１２０に通知するために、ノードピッカー１３０は、例えば、操作状態及び利用可能な供給源に関して、ノード１６０の状態を監視するように設定されてもよい。一実施形態において、ノードピッカー１３０は、現在操作可能なストレージサービスシステム内のノード１６０を識別するために、ＤＦＤＤ１１０（以下に説明する）のインスタンスと交流しあうように構成されてもよい。ノードピッカー１３０が操作可能ノード１６０に気付くと、これらのノードに、それぞれが利用可能な供給源（例えば、ストレージ容量）を確実にするためにクエリを行ってもよい。ノード１６０の操作状態及び供給源状態は時間と共に変化してもよいため、一部の実施形態において、ノードピッカー１３０はしばしばＤＦＤＤ１１０を介し、操作状態の情報を新しくし、これに伴うノード１６０がこの供給源状態情報を新しくすることを集計してもよい。一部の例において、ノードピッカー１３０は、ノード１６０の状態の完全な同期視野を有しない場合があることに留意されたい。例えば、ノードピッカー１３０によって使用可能であると思われる特定のノード１６０は、実際には状態情報の最近の更新のために機能しない場合がある。このような例において、ノードピッカー１３０は、コーディネータ１２０によって完了されることが可能である場合がある読み込み、又は書き込みプランを保証することが不可能である場合がある。コーディネータ１２０が、ノードピッカー１３０によって特定されるノード１６０へアクセスできない場合、関連の操作は機能しない場合があり、そっくりそのままコーディネータ１２０によって再度試みられるか、又はコーディネータ１２０は、ノードピッカー１３０が要求されたプランを改訂することを交渉してもよい。一部の事例において、義務的規定を満たすことが以前可能であるが、書き込みプランにおいて特定されるノード１６０の障害が、任意の又は最善努力のストレージ規定のみに影響を及ぼす場合、書き込みプランを完了することが可能であってもよい。このような一部の実施形態において、レプリケータ１８０は、以下に

説明するように、暫く経ってから満たされなかったストレージ規定を満たすための試みをするように設定されてもよい。

【００５０】

一部の実施形態において、ノードピッカー１３０の複数のインスタンスは、ストレージサービスシステムにわたって分散されてもよい。例えば、ノードピッカー１３０のそれぞれのインスタンスは、コーディネータ１２０の各インスタンス用に分散してもよい。ノードピッカー１３０は、ＡＰＩを介し、コーディネータ１２０（及び、レプリケータ１８０）からアクセスされてもよいサービスとして分散されてもよく、この構造は必須ではない。他の実施形態において、ノードピッカー１３０の機能性は、コーディネータ１２０及び／又はレプリケータ１８０のインスタンス内に直接合体されてもよい。

【００５１】

上記に記載されるように、オブジェクトデータの信頼性及び利用可能性は、ストレージデバイスシステムにわたるオブジェクト３０を複製することによって増加してもよい。例えば、地理的に分散されたシステム内でオブジェクト３０のインスタンス又は複製を分散することは、当該クライアントに近い一部のオブジェクトインスタンスを利用可能に配置することによって、当該オブジェクト３０へアクセスを試みる同様に分散されたクライアント５０の機能を改善してもよい。（オブジェクトの複製の内容において、「インスタンス」及び「複製」の用語は本書において同義的に使用される場合があることに留意されたい。）さらに、オブジェクト複製は、特定のオブジェクトインスタンスの破壊に起因するデータ損失の可能性を概して減少してもよい。しかしながら、所定の時における一部の実施形態においては、オブジェクト３０の有効な複製の数は、複製の所望の数又は目標とする数よりも少なくてもよい。例えば、ストレージサービスシステムに全体にわたり複製ストレージ規定を実行することは、各オブジェクト３０の複製の特定の目標数（例えば、又は他の適切な数）がいかなる所定の時においても存在することを特定してもよい。しかしながら、所定のオブジェクト３０について、有効な複製の実際の数は、様々な理由で目標の数より少ない場合がある。例えば、これまで有効な複製は、それが保存されたデバイスの障害のため、アクセス不可能になる場合がある。代わりに、一部の実施形態において、コーディネータ１２０によって書き込まれたオブジェクト３０のインスタンスの数は、そのオブジェクト３０のための複製の目標の数よりも少なくてもよい。例えば、上記に記載されるように、インスタンスは、ノードピッカー１３０によって特定される書き込みプランに従い書き込まれてもよく、目標の数よりも少ないインスタンスを要求する持続性規定を考慮に入れてもよい。

【００５２】

一実施形態において、レプリケータ１８０は、各オブジェクト３０の有効な複製の数が目標の数を満たすかどうか（例えば、判断が下される時点において、複製の数が、少なくとも目標の数であるかどうか）決定するために、オブジェクト３０を調べるために操作されてもよい。具体的には、一実施形態において、レプリケータ１８０は、各オブジェクト３０のインスタンスの数及び場所を特定する記録に渡り、継続して反復するように設定されてもよい。例えば、レプリケータ１８０は、以下に詳しく説明するキーマップインスタンス１４０のように、オブジェクトキーと複製されたオブジェクトインスタンスを識別する対応するロケータ間のマッピングを保存するように設定されてもよい、レプリケータキーマップ１９０を参照してもよい。（他の実施形態において、レプリケータ１８０は、キーマップの専用インスタンスよりは、１つのキーマップインスタンス１４０を参照してもよい。）一部の実施形態において、複数のレプリケータ１８０は、キーマップ空間の異なる部分を同時に調査するように設定されてもよく、ストレージサービスシステムによって管理されるすべてのオブジェクト３０の状態を調べるために必要とされる全体の時間を削減してもよい。

【００５３】

レプリケータ１８０が、所定のオブジェクト３０に対する有効な複製の目標とする数が満たされないと判断する場合、所定のオブジェクト３０への書き込み操作を行うコーディ

ネータ１２０と同様の方法で、レプリケータは所定のオブジェクト３０の追加的な複製を書き込むように設定されてもよい。例えば、レプリケータ１８０は、上記に記載されるように、付加的レプリケータの作成のための書き込みプランを取得するためにノードピッカー１３０とインターフェースで接続してもよい。代わりに、レプリケータ１８０は、オブジェクト複製を作成するための規定を反映する、独自のアルゴリズムを実行してもよい。一部の実施形態において、レプリケータ１８０は、付加的な複製が要求される状態下によって、オブジェクト３０のための複製を作成する異なる優先事項と一致してもよい。例えば、レプリケータキーマップ１９０にリストされたロケータの目標の数よりも少ない数を有するオブジェクト３０は、コーディネータ１２０によって最近書き込まれた場合がある。反対に、いくつかが無効であるロケータの目標とする数を有するオブジェクト３０は、根本的なストレージの障害を提示している場合がある。規定の問題として、レプリケータ１８０は、後者の事例の前に前者の事例を、訂正することを試みてもよく、その逆もまた同様である。代わりに、レプリケータ１８０は、該状態を生じさせる特定の状況に関わらず、この状態に遭遇する場合はいつでも、有効な複製の目標とする数よりも少ない数を有するあらゆるオブジェクト３０のための付加的な複製を作成することを試みてもよい。

【００５４】

上記に記載されるように、オブジェクト３０の全体的な信頼性は、例えば、異なる領域又はデータセンタ内に、オブジェクトデータの複製を保存することによって増加してもよい。しかしながら、一部の実施形態において、各複製は、オブジェクトデータの正確なコピーと対応する必要がないことに留意されたい。一実施形態において、オブジェクト３０は、オブジェクトデータが、すべての作成された部分を使用しなくても再度作成できる、冗長するコード化スキーム（パリティ、エラー訂正又は他のスキームなど）に従い多くの部分又は「破片」に分割されてもよい。例えば、オブジェクト３０からＮ部分を作成するために様々なスキームを使用することで、オブジェクトデータは、コード化スキームに従い、部分の任意のＮ－１、Ｎ部分の任意の過半数、又は部分の他の組み合せから再度作成されてもよい。このような実施形態において、オブジェクト３０の複製は、作成された部分、又は部分の一部の組み合せと対応してもよい。このような方法は、オブジェクトデータの複数の完成したコピーを保存することに比べ、データストレージ要求を減少すると同時に、効果的なフォルトトレランスを提供してもよい。しかしながら、一部の実施形態において、冗長するコード化技術は、オブジェクトデータの完成した複製との組み合せにおいても使用される場合があることに留意されたい。例えば、オブジェクトデータの複数の個々の完成したコピーは、上記に記載されるように、適切な冗長のコード化技術に従い、決定された複数部分のそれぞれの収集物としてノード１６０間に保存されてもよい。最後に、一部の実施形態において、一部のオブジェクト３０は、複製又はフォルトトレランスのいかなる度合いをも有して保存される必要がないことに留意されたい。例えば、ストレージクラスの説明とともに以下に説明するように、クライアントは、フォルトトレランスの高い程度を特定するストレージクラスについておそらく低コストである、ある程度の又は全く程度を有しないフォルトトレランスを特定するストレージクラスに従いオブジェクト３０が保存されることを要求してもよい。

【００５５】

一般的に言えば、キーマップインスタンス１４０は、オブジェクト３０のキーと特定のインスタンスのロケータ、又はオブジェクト３０の複製間の関係の記録を提供してもよい。このような記録を保存することにおいては、キーマップインスタンス１４０は、オブジェクト３０がストレージシステム内に複製された度合いも反映する（例えば、いくつのオブジェクト３０のインスタンスが存在するか、どのようにそれらが参照される場合があるか）。ビットストアノード１６０は、概して、ロケータによって識別されるオブジェクト３０の個々のインスタンスのためのストレージを提供してもよい。しかしながら。所定のノード１６０は、あらゆる他のノード１６０に関するインスタンスの状態、又は対応するオブジェクト３０のインスタンスのロケータ及びキーとの間の関係に気付かないことがある。すなわち、一般的に言うと、キーマップインスタンス１４０によって維持される状態

情報は、ビットストアノード１６０に透過的であることがある。ＤＦＤＤ１１０は、コーディネータ１２０及びレプリケータ１８０などのＤＦＤＤ１１０のクライアントが正確であるが、遅延の可能性が一部の検出された状態のビューを得るために、ノード１６０及び／又はキーマップインスタンス１４０（及び、実行されている場合、レプリケータキーマップ１９０）の操作状態についての状態情報を検出し通信する操作をしてもよい。これらのコンポーネントは、以下にさらに詳しく取り扱う。

【００５６】

図２のストレージサービスシステムアーキテクチャの一部のコンポーネントの物理的展開を示す一実施形態を、図３に示す。図示した実施形態において、データセンタ３００は、２つの領域である３１０ａ－ｂを含んで示される。さらに、領域３１０ｃ－ｄがデータセンタ３００の外部から示され、領域３１０ａ－ｄは、ネットワーク６０を介して相互接続されている。領域３１０ａ－ｄのそれぞれは、それぞれコーディネータインスタンス１２０ａ－ｄを含む。さらに領域３１０ａ－ｄは、図３に示されない、図２の他のコンポーネントのみならず、様々な組み合せのビットストアノード１６０及びキーマップインスタンス１４０を含んでもよい。例えば、領域３１０ａは４つのビットストアノード１６０を含み、領域３１０ｂは３つのビットストアノード１６０及びキーマップインスタンス１４０を含み、領域３１０ｃは２つのビットストアノード１６０を含み、領域３１０ｄは１つのビットストアノード１６０及び１つのキーマップインスタンス１４０を含む。

【００５７】

上記に記載されるように、一実施形態において、領域３１０ａ－ｄのそれぞれは、障害から独立している、又はほとんど相関しない場所と考えられてもよい。すなわち、障害が起こっているいかなる所定の領域３１０の可能性は、いかなる他の所定の領域３１０の障害の可能性から独立している、又は相関しないか、又は障害の可能性の相関性はしきい値量よりも低い場合がある。例えば、２つの領域３１０は、同時に障害をきたす可能性が１０％以下を示す。障害の相関関係又は独立性は、あらゆる適切な統計的又は確率的な技術を使用して測定され、様々な方法で実行されてもよい。例えば、領域３１０は、物理的に分割されるか、又は独立したユーティリティーグリッドへ接続されてもよく、これはおそらく１つの領域３１０に影響をおよぼす大災害が、他に影響を及ぼさないようにする。同様に、データセンタ３００内で、相異なる領域３１０は、１つの領域３１０が別の領域３１０の障害に関わらず、操作を継続することを可能にするために機能してもよい、独立したバックアップ電力源、ネットワーク接続、又は他の冗長性供給源を有してもよい。

【００５８】

一部の実施形態において、それぞれの障害の可能性間における、小さいがゼロではない相関間関係を有する２つの領域３１０は、障害の独立した可能性を有するとして依然と言われることもあることに留意されたい。例えば、バックアップ電力源、冷却などのためのしっかりとした、独立したシステムをそれぞれ有するにもかかわらず、所定のデータセンタ３００内の２つの領域３１０は、非常に大規模の大災害（例えば、データセンタ３００の全体を破壊するに十分な爆発）の場合における同時障害の影響を受けやすい場合がある。しかしながら、これらの２つの領域３１０が同時に機能しなくなるのに十分である事象の可能性は、非常に少なく、実用的な目的として、該２つの領域３１０は、障害の独立した可能性を有すると言える場合がある。

【００５９】

領域３１０は、階層の付加的レベルを含んでもよい（本書に示されない）。例えば、一実施形態において、あらゆる適切な領域体系が用いられる場合があるが、領域３１０は、ビットストアノード１６０などの、個々のノードにさらに細分化され、ラックに細分化されてもよい。一般的に言えば、領域３１０は、該領域内で展開されるストレージサービスコンポーネントを実装するために十分なコンピューティング供給源を含んでもよい。例えば、各ビットストアノード１６０は、図４から図９の説明とともに以下に説明するような様々なハードウェア及びソフトウェアコンポーネントを含む場合が一部の独立したコンピュータシステムとして実装されてもよい。同様に、各キーマップインスタンス１４０は、

図１０から図２２の説明とともに以下に説明するような構成の多くのコンピュータシステムを介して実装されてもよい。

【００６０】

一部の実施形態において、ウェブサービスプラットフォーム１００、コーディネータ１２０、ノードピッカー１３０、レプリケータ１８０、及びＤＦＤＤ１１０などのコンポーネントは、コンポーネントが展開される各領域３１０内の個別のコンピューティング供給源を介して実装されてもよい。例えば、これらのコンポーネントのそれぞれは、それぞれの計算システムによって実行可能なインストラクション及びデータの１組として実装されてもよい。代わりに、これらのコンポーネントのいくつか、又はすべては、１つ以上の計算システムで同時に実行する場合が一部の過程として実装されてもよい。一部の実施形態において、これらのコンポーネントのいくつか、又はすべてを実装するために使用されるコンピューティング供給源は、ビットストアノード１６０又はキーマップインスタンス１４０を実装するために使用されるこれらの供給源を共有してもよい。例えば、計算システムは、コーディネータ１２０機能性のみならず、キーマップ１４０機能性の一部分の、両方を実装するように設定されてもよい。一般的に言えば、個々の領域３１０内で展開されるコンピューティング供給源にわたる図２のコンポーネントのあらゆる適切なセグメント化が、用いられる場合がある。図３に示すように、異なる領域３１０は、ストレージシステムコンポーネントの異なる組み合せを含む場合があり、示された実施形態は、制限することよりもむしろ実例となることを意図したものであることに留意されたい。

【００６１】

さらに、異なるストレージサービスシステムコンポーネントは、通信プロトコルのあらゆる適切な種類に従い、通信してもよい。例えば、図２の一部のコンポーネントが個別のアプリケーション又は実行可能な過程として実装される時、これらはオペレーティングシステム、又はプラットフォーム（例えば、遠隔手続きコール、キュー、メールボックス、ソケットなど）によって、提供される場合がある標準のプロセス間の通信を互いに使用して、又は標準の、又は独自のプラットフォーム独立型通信プロトコルの使用によって通信してもよい。当該プロトコルは、ハンドシェイク／肯定応答、エラー検出及び訂正、又は通信コンポーネントに要求される、又は望まれる場合があるその他の通信特徴の任意のレベルをサポートしてもよい、処理状態を把握する、又は処理状態を把握しないプロトコルを含んでもよい。例えば、１つのストレージサービスシステムの実施形態において、コンポーネント間通信の実質的程度は、Ｔｒａｎｓｍｉｓｓｉｏｎ Ｃｏｎｔｒｏｌ Ｐｒｏｔｏｃｏｌ（ＴＣＰ）、Ｕｓｅｒ Ｄａｔａｇｒａｍ Ｐｒｏｔｏｃｏｌ（ＵＤＰ）、又は同様の標準の、又は独自の輸送プロトコルのバージョンなどの、適切なインターネット輸送層プロトコルを使用して実装されてもよい。しかしながら、ストレージサービスシステムコンポーネント間の通信は、プロトコル抽象化のより高い層にプロトコルを使用して実装される場合があることも考慮にされたい。例えば、クライアント５０とウェブサービスインターフェース１００間の通信のように、ストレージサービスシステムコンポーネント間の通信は、例えば、ＨＴＴＰ上のウェブサービスコールのような、アプリケーション層プロトコルを使用して実行されてもよい。

【００６２】

ビットストア構造

上記に論じるように、図２に示すストレージサービスシステムアーキテクチャにおいて、ビットストアノード１６０は、概してストレージサービスシステムによって管理されるオブジェクト３０に対するストレージを提供するために動作してもよい。ビットストアノード１６０の１つの例示的な実施形態を図４に示す。図で示された実施形態において、ビットストアノード１６０は、ストレージリパッカー１６３及び論理ファイルインプット／アウトプット（Ｉ／Ｏ）マネージャー１６５とインターフェースで接続するように構成されるストレージノード管理（ＳＮＭ）コントローラ１６１を含む。マネージャー１６５は、ファイルシステム１６７とインターフェースで接続するように構成され、入れ替わりに１つ以上のストレージデバイス１６９を管理するように構成される。様々な実施形態にお

いて、コントローラ１６１、ストレージレパッカー１６３、論理ファイルＩ／Ｏマネージャー１６５、又はファイルシステム１６７のいずれかは、コンピュータ接続可能媒体に保存され、以下に説明する機能を実行するためにコンピュータによって実行可能であってもよいインストラクションとして実装されてもよい。代わりに、これらのコンポーネントのいずれかは、専用のハードウェアサーキット、又はデバイスによって実装されてもよい。

【００６３】

一実施形態において、ＳＮＭコントローラ１６１は、ＡＰＩに従い、活動を実現するためにノード１６０の他のコンポーネントの活動を協調するのみならず、ノード１６０のクライアントに対するオブジェクトストレージＡＰＩを提供するように設定されてもよい。例えば、コントローラ１２０は、ＳＮＭコントローラ１６１によってもたらされるＡＰＩを介し、所定のノード１６０へのオブジェクト３０の保存する、及び所定のノード１６０からのオブジェクト３０の取り出すように設定されてもよい。ＡＰＩ管理が本願でＳＮＭコントローラ１６１の特徴として説明されるが、一部の実施形態において、ノード１６０のＡＰＩ処理機能はモジュールに、あるいは、ＳＮＭコントローラ１６１から相異なるコンポーネントに実装される場合があることを考慮されたい。

【００６４】

オブジェクトストレージＡＰＩは、オブジェクトを格納、取得、及び解放する操作をサポートしてもよい。１つのこのような実施形態において、総称的に保存操作、又は書き込み操作といわれることもある、オブジェクト格納操作は、操作の引数又はパラメータとしてオブジェクト３０のデータ及び／又はメタデータを特定してもよい。所定のノード１６０が完了次第、以下にさらに詳しく説明するように、ストレージサービスシステム全体に渡って保存される他のオブジェクト３０のすべてに対して取り出す所定のノード１６０のオブジェクトインスタンスを個別に識別してもよい、格納操作は、保存されたオブジェクト３０と対応するクライアントロケータを要求するクライアントに返してもよい。

【００６５】

反対に、総称的に読み込み、又は取り出し操作と言われることもある、オブジェクト取得操作は、パラメータとして、オブジェクト３０のロケータを特定してもよい。完了次第、取得操作は、特定されたロケータと対応するオブジェクトデータ及び／又はメタデータを、要求するクライアントに返してもよい。一部の実施形態において、該取得操作は、オブジェクトデータ、メタデータ、又はその両方がクライアントに対して返されるべきかどうかを要求するクライアントが特定することが出来るパラメータをサポートしてもよい。

【００６６】

取得操作のように、総称的に削除、又は除去操作と言われることもある、オブジェクト解放操作は、パラメータとしてオブジェクト３０のロケータを特定してもよい。しかしながら、完了次第、解放操作は、参照されるオブジェクト３０とあらかじめ関連してストレージリソースを解放してもよく、当該供給源は他のオブジェクト３０を保存するために使用されてもよい。一実施形態において、ロケータが解放されると、ロケータに対するそれに続く取得操作は一定期間続くか、又は続かなくてもよい。すなわち、解放操作は、再使用のためのストレージリソースを解放する場合が一部のノード１６０に対する信号としての役割を示す場合があるが、ノード１６０がそうすることを直ちに試みない、また通知を試みない場合には、クライアントとの当該の再使用を同期化してもよい。したがって、解放に続くオブジェクト３０へのアクセスのためのクライアントによる持続的な試みは、任意の期間にわたり継続される場合があり、それに続いてオブジェクト３０は予告無しにアクセス不可能になる場合がある。他の実施形態において、ノード１６０はオブジェクトデータが依然として使用可能であるか否かに関わらず、あらかじめ解放されたロケータへのクライアントのアクセスを妨げるように設定されてもよい。

【００６７】

様々な実施形態において、取得及び解放操作は、あらゆる適切なプロトコルに従い、他のパラメータを使用する及び／又は様々な状態、エラー又は他の表示を返すことがあることに考慮されたい。例えば、格納操作は、保存されるべき要求されたオブジェクト３０に

対するノード１６０に不十分な供給源がある場合、あるいは、何らかの他の理由で格納が完了されない場合、エラー状態を返すことがある。一部の実施形態において、ノード１６０のオブジェクトストレージＡＰＩは、他の操作を含んでもよいことを考慮されたい。例えば、ＡＰＩは、複製操作をサポートすることによってオブジェクト複製の作成を容易にするように設定されてもよい。一実施形態において、目的とするノード１６０に保存すべきオブジェクト３０のデータを供給する代わりに、要求するクライアントが異なるノード１６０にそのオブジェクト３０のロケータを特定することを除き、複製操作は、格納操作と同様に行われてもよい。目標とするノード１６０は、オブジェクトデータ及び／又はメタデータを得るために、特定されたノード１６０と交流し、目標とするノードと関連するオブジェクトのロケータをクライアントに返してもよい。他の実施形態において、ノード１６０は、オブジェクト３０での他の適切な操作をサポートしてもよい。

【００６８】

上記に記載されるように、格納、取得、及び解放操作を実装する、一部の実施形態において、存在するオブジェクト３０は同一の場所で改訂されない場合があることに留意されたい。むしろ、オブジェクト３０のインスタンスは、修正されたデータを含む新規インスタンスを書き込んだ後に、存在するインスタンスを解放することによって効果的に修正されてもよい。このような手段は、例えば、オブジェクト３０への修正のレンダリングが元のサイズよりも小さい、又は大きいことが生じた場合、フラグメンテーションの削減又はオブジェクト再配置によって、ノード１６０の根本的な管理層の実装を簡素化してもよい。ウェブサービスプラットフォーム１００に関して以下にさらに詳しく説明するように、一部の実施形態において、ストレージサービスシステムは、ラージオブジェクトをチャンクに分割し、各チャンクは相異なるオブジェクト３０として管理されてもよい。この手段は、再書込みが必要なことがあるチャンクの範囲を制限することによって頻繁に修正されてもよいラージオブジェクトを処理することにおいて、ノード１６０の機能を改善してもよい。しかしながら、他の実施形態において、ノード１６０は、今説明した再書込みの解放手段を通じるよりはむしろ、所定の位置に、オブジェクト３０の修正をサポートするために必要なこれらの特徴を含んでもよいことを考慮されたい。

【００６９】

図示した実施形態において、論理ファイルＩ／Ｏマネージャー１６５（又は、単にマネージャー１６５）は、オブジェクト３０が備わっている場合がある１つ以上の論理的連続ストレージ空間を、ＳＮＭコントローラ１６１及びリパッカー１６３にもたらすために、根本的なデバイス、又はファイルシステム特徴に仮想化するように設定されてもよい。例えば、所定のオブジェクト３０は、オフセットからのストレージ空間内及び範囲内のオフセット（例えば、データ及びメタデータを含むオブジェクトサイズに関して）に従い、論理ストレージ空間内に位置されてもよい。そのような論理ストレージ空間を提供することによって、マネージャー１６５は、当該根本的ストレージの実装の詳細に関わらず、ＳＮＭコントローラ１６１に対する根本的なストレージの均一の視野をもたらす場合がある。

【００７０】

論理ストレージ空間内でオブジェクト３０へアクセスすることを容易にするために、一実施形態において、マネージャー１６５は、各オブジェクト３０をノード１６０に保存するためのオブジェクトインデックス値（オブジェクトインデックスとしても言及される）を指定するように設定されてもよい。一般的に言えば、あらゆる所定のオブジェクト３０のインデックスは、所定のノード１６０内で固有であってもよい。例えば、一実施形態において、オブジェクトインデックスは、オブジェクト３０がノード１６０に保存される場合には常に、カウンタを増加し、オブジェクトインデックスとしての結果として得られたカウンタ値によって得られてもよい。（複数のオブジェクト書き込み操作が同時に進行することができる実施形態において、カウンタ増加は、例えば、シリアライゼーションを通じて、オブジェクトインデックス値が、一定の予想可能な形に指定されることを確実にするために、同期化されてもよい。）６４ビットの符号なし整数などの、十分に大きいカウンタ値は、例えば、実用的な目的のために、各オブジェクト３０が固有のインデックス値

を指定されることを確実にしてもよい。このようなカウンタは、例えば、2^{64}オブジェクトが保存された後一周してもよく、その後前もって作成されたインデックス値を繰り返してもよい。しかしながら、カウンタが一周した後にノード１６０内に所定のインデックス値を前もって指定されたオブジェクト３０が依然として存在するということは極めて可能性が低いように、コリジョンの可能性は非常に低い。オブジェクトインデックスを指定するためのあらゆる他の適切な方法も用いられる場合があることに留意されたい。以下に説明するように、オブジェクトインデックス値は、特定のオブジェクト３０を参照するためのコーディネータ１２０、又はノード１６０の他のクライアントによって使用されるロケータ値を決定するために、ノード１６０の固有の識別子との組み合せで使用されてもよい。

【００７１】

マネージャー１６５は、オブジェクト３０が、オブジェクトアクセスを容易にする方法で論理ストレージ空間内に位置される場所についての情報を体系化するために、上記に記載される固有のオブジェクトインデックス値を使用するように設定されてもよい。例えば、図５の上部に示すように、一実施形態において、マネージャー１６５は、オブジェクトインデックス値を介してすぐにアクセスできるように体系化されるテーブル、又は同様のデータ構造を保存するように設定されてもよい。図示した実施形態において、インデックステーブル５００は、多くのエントリ５１０を含んでもよく、エントリのそれぞれは、オブジェクトインデックスフィールド、オフセットフィールド、オブジェクトサイズフィールド、メタデータサイズフィールド、及び周期的冗長検査（ＣＲＣ）フィールドを含む多くのフィールドを含んでもよい。図５の下部に示すように、いくつかの例示的なオブジェクト３０について、エントリ５１０のオフセットフィールドは、論理ストレージ空間内の対応するオブジェクト３０の開始位置を特定してもよく、オブジェクトサイズ及びメタデータサイズフィールドは、オフセット位置からオブジェクトデータ及びメタデータを拡張する程度を特定してもよい。図示した実施形態において、他の実施形態においてこの順序は反対になる場合があるが、オブジェクトデータは、オブジェクトメタデータの上位である。ＣＲＣフィールドは、周期的冗長検査アルゴリズム又は他の適切な種類のチェックサム、又はハッシュアルゴリズムの結果、保存されてもよい。ＣＲＣフィールドに最初に保存される値は、オブジェクト３０が最初にノード１６０に保存された時に計算されてもよい。それに続いて、オブジェクト３０がアクセスされると、同一のアルゴリズムがオブジェクトデータ及び又はメタデータ、及び保存されたＣＲＣフィールド値と比較した結果値に適応されてもよい。比較が不一致の結果になった場合、保存されたデータの整合性に障害が起きた可能性がある。他の実施形態において、エントリ５１０は、付加的な、又はこれら示されたものから異なるフィールドを含んでもよいことに留意されたい。例えば、ＣＲＣフィールドは省略されるか、又はどこかに実装されてもよい。さらに、オブジェクトデータ及びメタデータの固定位置は、関連のオフセットに加えて、又はそれに代わって保存されてもよい。

【００７２】

リパッカー１６３は、オブジェクト３０が解放される時、及びその関連するストレージリソースが再要求された時に現れるギャップを除くための論理オブジェクトストレージ空間で操作するように設定されてもよい。一実施形態において、リパッカー１６３は、前回の解放操作によって解放されているＳＮＭコントローラ１６１及び／又はマネージャー１６５によってマークされているオブジェクト３０を識別するために、論理オブジェクトストレージ空間（例えば、一定期間又は継続的に）スキャンするように設定されてもよい。リパッカー１６３は、解放されたオブジェクト３０の除去を反映するために更新すべき解放されたオブジェクト３０のインデックスの後に現れるインデックスを伴うこれらオブジェクト３０のエントリ５１０をもたらしてもよく、これらのオブジェクト３０が、論理オブジェクトストレージ空間の元の位置にむかって移動する効果的な結果となる場合がある。例えば、図５の下部のオブジェクトＮが解放された場合、リパッカー１６３は、オブジェクトＮ＋１の新規のオフセットフィールドとして、オブジェクトＮのオフセットフィー

ルドを反映するために更新されるべきオブジェクトＮ＋１と対応するエントリ５１０をもたらすために操作してもよい。リパッカー１６３は、削除されるべきオブジェクトＮに関連するエントリ５１０をもたらしてもよく、移動を反映するためにオブジェクトＮ＋１に続くオブジェクトのオフセットを更新してもよい。一実施形態において、マネージャー１６５は、オブェクトデータ及びメタデータの対応する移動を、論理オブジェクトストレージ空間及び／又はストレージデバイス１６９の根本を成すフィールド又は構造内におこさせるように動作してもよい。

【００７３】

一部の実施形態において、マネージャー１６５は、異なる種類のハードウェア及びソフトウェアを含む複数の異なる実行プラットフォーム上で実行するように設定されてもよい。このような一部の実施形態において、抽象概念の１つ以上の付加的な層は、ＳＮＭコントローラ１６１及びそのクライアントにマネージャー１６５によってもたらされる論理オブジェクトストレージ空間の間に存在してもよい。例えば、図示した実施形態において、マネージャー１６５は、ファイルシステム１６７によって管理される１つ以上の物理的ファイルとしての論理オブジェクトストレージ空間を実装するように設定されてもよい。一般的に言うと、ファイルシステム１６７は、様々な種類の物理的ストレージデバイス１６９を、本願において物理的ファイルとして言及される論理単位にデータを保存する論理ストレージデバイスに体系化するように設定されてもよい。ファイルシステム１６７によって管理される論理ストレージデバイスは、事実上、階層的であってもよい。例えば、ファイルシステム１６７は、物理的ファイルを保存し、それにアクセスするために誘導される場合があるディレクトリ、又はフォルダの階層をサポートしてもよい。一般的にいえば、ファイルシステム１６７は、所定の物理的ファイルと、物理的ファイルの対応するデータ及び／又はメタデータが保存されるストレージデバイス１６９の位置間の関係を追跡し、管理するように設定されてもよい。したがって、一実施形態において、マネージャー１６５は、ファイルシステム１６７によって配置される１つ以上の物理的ファイルに対する論理的オブジェクトストレージ空間のマッピングを管理してもよい。入れ替わりに、ファイルシステム１６７は、ストレージデバイス１６９のアドレス可能な位置に対するこれらの物理的ファイルのマッピングを管理してもよい。

【００７４】

根本を成すデバイス１６９の管理のための異なる特徴を提供する、あらゆる所定のオペレーティングシステムは、様々な異なるファイルシステム１６７をサポートしてもよいが、ファイルシステム１６７は、オペレーティングシステム内で概して統合されてもよい。例えば、様々なバージョンのＭｉｃｒｏｓｏｆｔＷｉｎｄｏｗｓ（登録商標）オペレーティングシステムは、ＦＡＴ３２（ＦｉｌｅＡｌｌｏｃａｔｉｏｎＴａｂｌｅ－３２）及びＦＡＴ１６ファイルシステムのみならず、ＮＴファイルシステム（ＮＴＦＳ）などのファイルシステムをサポートする。様々なバージョンのＬｉｎｕｘａｎｄＵｎｉｘ（登録商標）オペレーティングシステムは、ｅｘｔ／ｅｘｔ２ファイルシステム、ＮｅｔｗｏｒｋＦｉｌｅＳｙｓｔｅｍ（ＮＦＳ）、ＲｅｉｓｅｒＦｉｌｅＳｙｓｔｅｍ（ＲｅｉｓｅｒＦＳ）、ＦａｓｔＦｉｌｅＳｙｓｔｅｍ（ＦＦＳ）、その他多数のファイルシステムをサポートしてもよい。一部の第三者ソフトウェアベンダーは、例えば、ＶＥＲＩＴＡＳ（登録商標）ＦｉｌｅＳｙｓｔｅｍ（ＶｘＦＳ）などの様々なコンピューティングプラットフォームとの統合のための独自のファイルシステムを提供してもよい。異なるファイルシステムは、根本を成すストレージデバイス１６９の管理のため、様々な特徴のためのサポートを提供してもよい。例えば、一部のファイルシステム１６７は、デバイスミラーリング、ストライピング、スナップショッティング、又は他の種類の仮想化特徴のためのサポートを提供してもよい。

【００７５】

一部の実施形態において、マネージャー１６５とストレージデバイス１６９間に依然として抽象概念のさらなる層が存在してもよい。例えば、一部の実施形態において、ボリュームマネージャー層は、ファイルシステム１６７とストレージデバイス１６９の間に提供

されてもよく、上述の仮想化特長のいくつか、あるいはすべてを行うように設定されてもよい。代わりに、特定のストレージデバイス１６９は、ハードディスクドライブの独立した配列、又は仮想化コントローラを含む他のデバイスとして設定されてもよい。上記に記載されるように、ボリュームマネージャーによって、又はファイルシステム１６７内でサポートされてもよい仮想化マッピングと同様に、内部で、仮想化コントローラは、ディスクドライブへデバイスのストレージアドレス空間の複雑なマッピングを任意にサポートしてもよいが、仮想化コントローラは、単一の物理的デバイスとして、ファイルシステム１６７にディスクドライブをもたらすように構成されてもよい。一部の実施形態においては、示した抽象概念の層よりも少ない場合があることにも留意されたい。例えば、一部の実施形態において、マネージャー１６５は、ファイルシステム１６７を使用することなく、例えば、未加工の物理的デバイスとしてのストレージデバイス１６９と直接交流するように設定されてもよい。

【００７６】

一般的に言えば、ストレージデバイス１６９は、ファイルシステム１６７及び／又はマネージャー１６５によってサポートされてもよいストレージデバイスのあらゆる適切な種類を含んでもよい。ストレージデバイス１６９は、Ｓｍａｌｌ Ｃｏｍｐｕｔｅｒ Ｓｙｓｔｅｍ Ｉｎｔｅｒｆａｃｅ（ＳＣＳＩ）デバイス、又はＡＴ Ａｔｔａｃｈｍｅｎｔ ＰｒｏｇｒａｍｍｉｎｇＩｎｔｅｒｆａｃｅ（ＡＴＡＰＩ）デバイス（Ｉｎｔｅｇｒａｔｅｄ Ｄｒｉｖｅ Ｅｌｅｃｔｒｏｎｉｃｓ（ＩＤＥ）デバイスとしても知られる）ようなハードディスクドライブデバイスを一般に含んでもよい。しかしながら、ストレージデバイス１６９は、磁気、又は光学媒体系のデバイス、ソリッドステート大容量記憶装置（例えば、不揮発性の、又は「フラッシュ」メモリ系のデバイス）、磁気テープなどを含む、あらゆる種類の大容量記憶デバイスを網羅してもよい。さらに、ストレージデバイス１６９は、Ｕｎｉｖｅｒｓａｌ Ｓｅｒｉａｌ Ｂｕｓ又はＩＥＥＥ１３９４／Ｆｉｒｅｗｉｒｅ（登録商標）基準のバージョンに準拠するインターフェースなど、上述のものに加え、あらゆる適切なインターフェース種を通じてサポートされてもよい。

【００７７】

上記に記載されるように、ストレージサービスシステム内に保存されるオブジェクト３０のあらゆる所定のインスタンスについて、対応するロケータは、該システム内のすべてのノード１６０にわたるインスタンスを個別に識別してもよい。一実施形態において、ロケータは、個別の識別子、又はオブジェクトインスタンスが保存されるノード１６０と対応する、「ノードＩＤ」のみならず、マネージャー１６５によってオブジェクトに指定される、連結、組み合せ、又は、オブジェクトインデックス値の他の機能として作成されてもよい。例えば、上記に記載されるように、６４ビットオブジェクトインデックス値は、１２８ビットロケータを生じる６４ビットノードＩＤと組み合わせてもよい。様々な実施形態において、ロケータを形成するために、より少ない、又は大きい数のビットが用いられてもよいが、このようなロケータは、2^{64}と同数の固有ノード１６０が2^{64}と同数の固有のオブジェクトインスタンスを保存することを可能にする。

【００７８】

一実施形態において、ノードＩＤは、タイムスタンプ、又はデートスタンプを有する、所定のノード１６０と対応する、ＩｎｔｅｒｎｅｔＰｒｏｔｏｃｏｌ（ＩＰ）アドレスなどの、固有のネットワークアドレスの連結、又は結合を通じて形成されてもよい。例えば、ノード１６０は、ＩＰアドレスが指定された時点を反映するタイムスタンプと、又はＩＰアドレスが有効であるとわかっている間の時点との結合において、そのＩＰアドレスに従い（例えば、ノードスタートアップ／初期設定において、又はノードＩＤが指定された時点で、さもなければ、初期設定の間に）、ノードＩＤを指定されてもよい。一般的に言えば、同一のＩＰアドレス空間に所属する２つの相異なるノード１６０は、いかなる所定の時において、同一のＩＰアドレスを有効に指定されることはない。したがって、ノードのＩＰアドレスとタイムスタンプの値の結合は、そのノードに固有の識別子を生じてもよい。例えば、３２ビットのＩＰアドレスは、他のビット幅も用いられてもよいが、上述の

６４ビットノードＩＤを生じるために、３２ビットのタイムスタンプ（例えば、一部の共通の参照時間のため、それは経過した秒数を示す）と連結、又は結合されてもよい。ノードＩＰアドレスに頼らない固有のノードＩＤを指定するために、他の技術が使用されてもよいことを考慮されたい。例えば、ネームサーバなどの中央局は、上記に記載されるように、ノード１６０内のオブジェクトインデックス値の指定と同様に、ノードＩＤの固有性を保証する方法での要求におけるノードＩＤを委任してもよい。

【００７９】

ノードＩＤがノードＩＰアドレスから派生したノードＩＤで一部の実施形態において、該ノードＩＤは、いかなる時にもノード１６０の現在のＩＰアドレスを反映しなくてもよいことに留意されたい。例えば、ノードＩＤは、ノード１６０が初期化されるまで存続するが、ノードのＩＰアドレスは、ノードＩＤの作成に続いて変更されるか、又は再指定されてもよい。また、一部の実施形態において、ノードＩＤは、ストレージクライアント５０又は他の悪意のあるエンティティが実際のノードＩＰアドレスを決定するためにロケータを解読することから防ぐために、決定的な方法でハッシュされ、暗号化され、又は分かりにくくなる場合がある。

【００８０】

図４のノード１６０の実施形態に関する取得、格納、及び解放操作の例示的な実施形態の操作は、図６から図８に示される。最初から図６までに言及して、取得操作は、該操作がコーディネータ１２０又は他のクライアントからノード１６０で受信される、ブロック６００で始まってもよい。例えば、コーディネータ１２０は、上記に記載されるように、ノードＩＤ及びオブェクトインデックス値を含む特定のロケータに取得操作を発行してもよい。該ノードＩＤは、例えば、ノードＩＤが目標とするノード１６０の現在のＩＰアドレスを反映する場合、適切なノード１６０に対して取得操作を直接送るために使用されてもよい。代わりに、以下に説明するように、ＤＦＤＤ１１０などのディレクトリサービスが、ノードＩＤを、取得操作が適切なノード１６０に送られることを通じるアドレス可能なエンドポイント又は目的地に分解するために用いられてもよい。

【００８１】

ノード１６０によって受信されると、取得操作は、ノード１６０の論理オブジェクトストレージ空間内の目標とされるオブジェクトインスタンスの範囲を識別するために処理されてもよい（ブロック６０２）。例えば、コントローラ１６１は、取得操作を受信し、それをマネージャー１６５へ伝えてもよい。入れ替わりに、マネージャー１６５は、論理オブジェクトストレージ空間内の所望のオブジェクトインスタンスの位置を得るために取得操作がインデックステーブル５００にアクセスすることによって参照される、ロケータのオブジェクトインデックス部分を使用してもよい。例えば、マネージャー１６５は、オフセットからオブジェクトインスタンスの長さのみならず、オブジェクトインスタンスが開始する、論理オブジェクトストレージ空間にオフセットを得てもよい。一部の実施形態において、取得操作は、オブジェクトデータ、メタデータ、又はその両方が要望されるかを特定してもよい。そのような実施形態において、マネージャー１６５は、要求データと関連し、論理オブジェクトストレージ範囲を決定してもよい。例えば、オブジェクトデータ及びメタデータの両方が要望される場合、マネージャー１６５は、取り出すべきオブジェクトオフセットからの範囲を決定するために、オブジェクトデータサイズ及びメタデータサイズの両方を使用してもよい。上記に述べたように、他の実施形態において、オブジェクトインスタンスのためのストレージ範囲は、論理オブジェクトストレージ空間内の関連するオフセットよりは、固定位置を通じるなどして、異なる方法でマネージャー１６５によって保存、及び管理されてもよい。

【００８２】

論理オブジェクトストレージ空間内のオブジェクト範囲は、物理的ファイルストレージ空間内の１つ以上の対応するファイル内のエクステントにマップ化されてもよい（ブロック６０４）。例えば、マネージャー１６５は、論理オブジェクトストレージ空間を、ファイルシステム１６７によって管理される１つ以上のファイルにマップし、例えば、読み込

まれるべき指定されたファイル内のロケータ、又はオフセットのみならず、１つ以上のファイルネームを反映することによって、所望のオブジェクト範囲と対応するデータを得るために、ファイルシステム１６７に適切なファイルアクセス操作を発行してもよい。他の実施形態において、コントローラ１６１は、マネージャー１６５によって管理される論理ブロックストレージ空間を迂回するように設定されてもよく、代わりに、ファイルシステム１６７によって管理される物理ファイルと直接交流してもよいことを考慮されたい。

【００８３】

物理的ファイルへの参照は、デバイス関連要求（ブロック６０６）に対してマップ化されてもよい。例えば、ファイルシステム１６７は、論理ブロックアドレス（ＬＢＡ）又はデバイス形状（例えば、シリンダ、トラック、セクタ及び／又はヘッド）に特定のアドレスなど、ストレージデバイス１６９のアドレス可能な位置を特定するための１つ以上の読み込み要求を作成するように設定されてもよい。上記に述べたように、一部の実施形態において、マネージャー１６５は、ファイルシステム１６７を迂回し、ストレージデバイス１６９を直接管理するように設定されてもよい。

【００８４】

要求されたオブジェクトデータは、ストレージデバイス１６９（ブロック６０８）から取り出され、要求するクライアント（ブロック６１０）へ返されてもよい。例えば、取り出されたデータは、図４に階層的に示される要求を通じたバックアップを通過されてもよく、又は、要求するクライアントへの輸送のために、ストレージデバイス１６９、又はファイルシステム１６７からコントローラ１６１へ直接返されてもよい。

【００８５】

図７に示すように、一実施形態において、図６のブロック６００について上記の記載と同様の方法で、操作が、コーディネータ１２０、又は他のクライアントからノード１６０で受信される時、格納操作はブロック７００で開始されてもよい。例えば、コーディネータ１２０は、ノードピッカー１３０によって作成される書き込みプランにおいて特定されるノード１６０に対して格納操作を発行してもよい。取得操作とは対照的に、格納操作は、保存されるべきオブジェクトデータ及び／又はメタデータを含んでもよく、任意に、データ及び／又はメタデータの長さを指定する付加的なパラメータを含んでもよい。

【００８６】

ノード１６０によって受信されると、格納操作は、論理オブジェクトストレージ空間内のオブジェクトインスタンスのためのストレージ範囲を指定するために処理されてもよい（ブロック７０２）。一実施形態において、マネージャー１６５は、オブジェクトインデックス値を新規オブジェクトインスタンスに指定し、インデックステーブル５００を新規オブジェクトインスタンスのオフセットを特定する新規エントリ５１０に記録するように設定されてもよい。例えば、新規エントリのオフセットは、インデックスの最高値を有する存在するオブジェクトインスタンスのストレージ範囲（例えば、オフセット及び長さ）に関連して決定されてもよい。新規オブジェクトインスタンスのデータ及び／又はメタデータの長さが格納操作に対するパラメータとして特定されない場合、マネージャー１６５、又はコントローラ１６１は、新規エントリ５１０における含有としてこれらを計算するように設定されてもよい。

【００８７】

論理オブジェクトストレージ空間内の新規に指定されたストレージ範囲は、物理的ファイルストレージ空間内の１つ以上の対応するファイル内のエクステントに対しマップ化されてもよい（ブロック７０４）。例えば、新規オブジェクトインスタンスに対して指定された範囲は、１つ以上の存在する物理的ファイルの最後に加えられてもよく、そうでない場合は、存在する、又は新規に割り振られた物理的ファイル内に位置されてもよい。物理ファイル範囲は、例えば、取得操作について上記の記載と同様の方法で、ファイルシステム１６７によってストレージデバイス範囲にマップ化されてもよく（ブロック７０６）、オブジェクトインスタンスデータ及び／又はメタデータは、ストレージデバイス１６９に保存されてもよい（ブロック７０８）。

【００８８】
データ及び／又はメタデータがうまくストレージデバイス１６９に書き込まれたことを確認すると、保存されたオブジェクトインスタンスに対応するロケータが、要求するクライアントに返されてもよい（ブロック７１０）。例えば、マネージャー１６５は、ノード１６０のノードＩＤに作成されたオブジェクトインデックス値を加えるように設定されてもよく、物理的ファイル書き込み操作が無事に完了するファイルシステム１６７からの表示に際しオブジェクトロケータとして結果値を返してもよい。
【００８９】
図８に示すように、一実施形態において、図６のブロック６００について上記の記載と同様の方法で、操作がコーディネータ１２０、又は他のクライアントからノード１６０で受信されるとき、解放操作は、ブロック８００で開始されてもよい。解放された操作は、他の実施形態において、他の引数も提供されるが、解放されるべきオブジェクトインスタンスのロケータを単に特定してもよい。
【００９０】
取得操作のように、ノード１６０によって受信されると、解放操作は、ノード１６０の論理オブジェクトストレージ空間内の目標とされるオブジェクトインスタンスの範囲を識別するために処理されてもよい（ブロック８０２）。例えば、コントローラ１６１は、解放操作を受信し、それをマネージャー１６５へ伝えてもよい。入れ替わりに、マネージャー１６５は、参照されるオブジェクトインスタンスの対応するエントリ５１０を識別するために、解放操作がインデックステーブル５００にアクセスすることによって参照される、ロケータのオブジェクトインデックス部分を使用してもよい。参照されるオブジェクトは、解放されたとしてマークされてもよい（ブロック８０４）。例えば、マネージャー１６５は、エントリがもはや有効ではないことを表す、負の数のような不正値に対し、オフセット、又は別のフィールドエントリ５１０を設定するように設定されてもよい。肯定応答が、オブジェクトが解放されたことを示す要求するクライアントに返されてもよい（ブロック８０６）。
【００９１】
上記に記載されるように、オブジェクトインスタンスと関連するストレージリソースは、オブジェクトインスタンスが解放された時に他の使用のために直ちに解放、再要求、又は再配分されてもよい。むしろ、一実施形態において、これらの供給源は、解放操作がこれらを再要求することに関し、非同期的に操作する独立過程まで存続してもよい。図９は、例えばストレージリパッカー１６３によって実装されてもよいような、当該の過程の一実施形態の操作を示す。ブロック９００において、ノード１６０に保存された、特定のオブジェクトインスタンスに対応するオブジェクトインデックスエントリが選択されてもよい。例えば、リパッカー１６３は、エントリに保存されたオブジェクトインデックス値に従い、配列の順序で、インデックステーブル５００からインデックスエントリ５１０を選択するように構成さてもよい。それに続いて、選択されたエントリは、対応するオブジェクトインスタンスが解放されたかどうかを決定するために調査されてもよい（ブロック９０２）。例えば、リパッカー１６３は、フィールドが、負の数値、又はある他の値などの、対応するオブジェクトインスタンスが解放されたことを示す値を設定したかどうかを確かめるために、オフセットフィールド、又は別のフィールドを調べてもよい。
【００９２】
選択されたオブジェクトが解放されていない場合、操作は、別のオブジェクトが選択されるブロック９００に戻って続行されてもよい。選択されたオブジェクトが解放された場合、論理オブジェクトストレージ空間は、解放されたオブジェクト（ブロック９０４）と対応するストレージリソースに再要求するために、再圧縮されてもよい。例えば、リパッカー１６３は、第１の当該オブジェクトインスタンスのオフセットが解放されたオブジェクトのオフセットに設定され、次の当該オブジェクトインスタンスのオフセットがデータサイズの機能として設定され、メタデータサイズと第１の当該オブジェクトインスタンスのオフセット、など、論理オブジェクトストレージ空間内の解放されたオブジェクトに続

く、これらのオブジェクトインスタンスのインデックスエントリ５１０を調整するように設定されてもよい。しかしながら、一部の実施形態において、解放されたオブジェクトインスタンスに続くオブジェクトインスタンスのすべてが、新規オブジェクトが調査のために選択される前に再圧縮される必要があるわけではない。例えば、再圧縮は、接触する各オブジェクトが、それが調査のために選択される時に圧縮されるように、オブジェクト選択と交互配置してもよい。

【００９３】

一部の実施形態において、マネージャー１６５は、論理オブジェクトストレージ空間の再圧縮に応じて物理的ファイルストレージ空間内で、同様の再圧縮、又は圧密操作を行ってもよい。例えば、マネージャー１６５は、論理オブジェクトデータ範囲が、異なる物理的ファイルデータ範囲に再マップ化されることを生じてもよい。同様に、一部の実施形態において、ファイルシステム１６７は、物理的ファイルストレージ空間の再圧縮に応じてストレージデバイス１６９間で、類似の再圧縮、又は圧密操作を行ってもよい。他の実施形態において、物理的ファイルストレージ空間、又はストレージデバイスそれ自体の再圧縮は、リパッカー１６３によって開始される論理オブェクトストレージ空間再圧縮を独立して発生してもよい。例えば、ファイルシステム１６７は、マップ化されたデバイスストレージエクステントが、ストレージデバイスのアクセス傾向に関連して、ほとんど、又は完全に連続的であるような、デバイスストレージエクステントへの物理的ファイルストレージエクステントのマッピングを再配置することによって、ストレージデバイスに保存された物理的ファイルを最適化するように構成されてもよい。

【００９４】

論理オブジェクトストレージ空間の圧縮に続き、解放されたオブジェクトに対応するインデックスエントリが削除され（ブロック９０６）、操作は、別のオブジェクトが選択されるブロック９００から継続されてもよい。上記に述べたように、一部の実施形態において、複数オブジェクトが選択された場合「オンザフライ」で再圧縮が発生してもよく、オブジェクトを再配置するために要求される操作の数を最小限にするため、論理オブジェクトストレージ空間の全体的な稼働率を改善してもよい。

【００９５】

一部の実施形態において、ノード１６０によってサポートされてもよい取得、格納、解放、又は他の操作のいずれかは、要求するクライアントに関する様々な種類のハンドシェイク、肯定応答、又はエラー処理プロトコルをサポートしてもよいことに留意されたい。例えば、クライアントが操作に対し、不正な形式での要求を要求する場合（例えば、必要なパラメータを供給することを怠る）、又はノード１６０が、操作を十分に完了できない場合（例えば、それが格納操作を引き受けるのに不十分な供給源を有する）、ノード１６０が要求するクライアントにエラー表示を返してもよい。このような表示は障害状態の性質のため、特定の詳細を含む、又は含まなくてもよい。

【００９６】

一実施形態において、コーディネータ１２０は、複数の操作が共通のデータを有してもよいが、操作によって目的とされる各それぞれのノード１６０に対する操作を独立して伝えるように設定されてもよい。例えば、書き込みプランに従い、複数のノード１６０にオブジェクト３０が書き込まれる格納操作の事例において、コーディネータ１２０は、各特定のノード１６０と独立して通信してもよい。しかしながら、他の実施形態において、複数目的地ノード１６０に意図された共通データ及び／又はパラメータを有する操作は、連鎖されてもよい。一実施形態において、コーディネータ１２０、又は他のクライアントは、受信者リストなど、操作のパラメータにおける各受信者を特定することによって、連鎖操作を先導してもよい。操作において示唆される複数の受信者は、初期設定によって連鎖を示してもよく、又は別のパラメータが、操作が連鎖されていることをマークするために使用されてもよい。コーディネータ１２０、又は他のクライアントが、操作に特定される最初の目的地ノード１６０の１つに連鎖操作を伝えることによって先導してもよい。

【００９７】

連鎖操作を受け取る際に、ノード１６０は、操作を処理し、それを操作において特定される別の目的地ノード１６０の１つに転送してもよい。当該の転送に先立ち、受け手ノード１６０は、受け手を示し、循環転送を避けるために、操作に含まれる目的地リストからそれ自体を取り除いてもよい。操作は、受け手ノードの処理と同時に転送されてもよい。代わりに、転送は、受け手ノードの無事な処理の完了を条件としてもよい。一部の実施形態において、連鎖操作は、受信者が操作内で示される順序で受信者に伝えられてもよい。他の実施形態において、ノード１６０は、例えば、残っている目的地のどれが一番近いか、最低負荷であるか、又は、いくつかの他の選択基準を満たすかを決定することによって、動的に次の受信者を選択してもよい。一部の実施形態において、連鎖、及び非連鎖操作の組み合せは、コーディネータ１２０、又は他のクライアントによって作成されてもよいことに留意されたい。例えば、同一のデータが、６つの相異なるノード１６０行きである格納操作の目標である場合、コーディネータ１２０は、６つの目的地ノード、又はそれぞれが３つの目的地ノードを特定する２つの連鎖操作を特定する、単一連鎖操作を作成してもよい。コーディネータ１２０が、各それぞれの目的地ノード１６０に独立して伝える、６つの非連鎖操作の作成を含む、他の組み合せも可能である。

【００９８】

キーマップ構造

上記に記載されるように、様々なビットストアノード１６０は、オブジェクト３０のインスタンスのためのストレージを提供するように設定されてよい。ノード１６０は、個別に冗長、又はデータセキュリティのためのあらゆる特定のサポートを提供しなくてもよく、実際に、一部の実施形態において、ノード１６０は、オープンソースのオペレーティングシステム（例えば、Ｌｉｎｕｘ）を作動する包括的なコンピュータプラットフォームを使用し、安価な、商品ハードドライブ（例えば、ＡＴＡＰＩ／ＤＥハードドライブ）を介してストレージを提供することで実装されてもよい。このような実施形態において、個々のシステムは特にフォルトトレラントでなくてもよい。むしろ、データセキュリティ及び冗長は、上記に記載されるように、多くのノード１６０に渡るオブジェクト３０の複製を通じて提供されてもよい。

【００９９】

前に論述したように、所定のオブジェクト３０は、ストレージクライアントによって特定されてもよいキーと対応してもよい。所定のオブジェクト３０の個々のインスタンスは、ストレージサービスシステム内に含まれるノード１６０の収集物にわたり、これらインスタンスを固有に識別してもよいそれぞれのロケータに対応してもよい。一実施形態において、ストレージサービスシステム内に展開される各キーマップインスタンス１４０は、所定のオブジェクト３０のためのキー及びすべての対応するロケータと、ノード１６０間に保存されたその複製されたインスタンスとの間の関係、又はマップを保存し、維持するように設定されてもよい。以下の論考において、どのようにキーマップインスタンス１４０の特定の実施形態が実装されてもよいかの説明に続き、キーマップインスタンス１４０の様々な実施形態の一般的な特徴及び機能性を論述する。

【０１００】

一実施形態において、所定のキーマップインスタンス１４０は、１つ以上のテーブル、又はあらゆる他の適切な種類のデータ構造内の様々なキーと関連するロケータ間の関係の詳細を保存するように設定されてもよい。例えば、図１０に示す一実施形態において、キーマップインスタンス１４０は、多くのエントリ１４４を有するキーマップデータ構造１４２を含む。各エントリは、関連する記録１４８のみならず、それぞれのキー１４６を含む。一部の実施形態において、以下にさらに詳しく説明するように、エントリ１４４を体系化するために使用されるデータ構造の体系は複雑であってもよい。しかしながら、機能的な見地から、キーマップインスタンス１４０は一般的に、所定のキー１４４とその対応する記録１４８との間に、１対１のテーブル型の関係を維持してもよい。

【０１０１】

記録１４８は、概して、所定のキー１４４と対応するロケータを含んでよいが、他の情

報を含んでもよい。例えば、記録１４８の一実施形態は以下のように設定されてもよい。

```
struct KeyRecord {
    int16_t version;
    int16_t storageClass;
    int64_t creationDate;
    int64_t objectSize;
    uint32_t crc32;
    int8_t numLocators;
    struct locator {
        int64_t nodeID;
        int64_t objectIndex;
    } replicas [];
}
```

このデータ構造例は、Ｃプログラミング言語の構文を使用して表現されるが、あらゆる適切な言語、表現、又は形式を使用して実装されてもよい。記録１４８の他の実施形態は、これら示されたものよりもさらに多い、さらに少ない、又は異なるフィールドを含んでもよい。一部の例では、記録１４８は、一部の種のＵｎｉｘファイルシステムに用いられるアイノード構造に対するストレージ空間の体系化における記録１４８の相似の目的で示す、「アイノード」といわれることもある。しかしながら、該文脈中の「アイノード」という用語の使用は、ファイルシステム、又は他のストレージ文脈内のアイノードの実装、又は使用の特定の詳細を引き合いに出すことを意図しない。

【０１０２】

上記の実施形態において、記録１４８は、７つの特定の要素を含む。１６ビットバージョンの要素は、記録１４８の形式に特定である、個別の識別値を保存するために使用されてもよい。例えば、異なる種類の記録１４８は、キーマップインスタンス１４０の異なる実装に使用されてもよく、一部の実施形態において、所定のキーマップインスタンス１４０内に保存される記録１４８は、異種であってもよい。バージョン要素は、記録１４８の異なるバージョン間で区別するために使用されてもよく、記録の他の要素が適切に解読され、使用されてもよい。

【０１０３】

１６ビットｓｔｏｒａｇｅＣｌａｓｓ要素は、記録１４８と対応するオブジェクト３０のストレージクラスの表示を保存するために使用されてもよい。ストレージクラスは、次の項にさらに詳しく述べる。一般的に言えば、オブジェクトの所定のストレージクラスは、所定のストレージクラスの他のメンバーに共通であってもよいストレージ特徴及び／又はポリシーを識別してもよいが、他のストレージクラスのメンバーからは異なってもよい。例えば、「高信頼性」ストレージクラス及び「低信頼性」ストレージクラスは、ストレージサービスシステムの所定の実装のために定義されてもよい。高信頼性ストレージクラスのメンバーであるオブジェクト３０は、低信頼性ストレージクラスのメンバーであるオブジェクト３０のより高い程度にたいして複製されてもよく、したがって、おそらく、低信頼性ストレージクラスのメンバーにたいして見積もられたものよりも高い使用コストと引き換えに、個々の複製の損失に対する敏感性を減少してもよい。多くの他の可能なストレージクラスの種類及び組み合せが可能であり、考慮される。

【０１０４】

６４ビットｃｒｅａｔｉｏｎＤａｔｅ要素は、対応するオブジェクト３０がストレージサービスシステム内に作成された日付及び時間の表示を保存するために使用されてもよい。この要素は、あらゆる適切な方法に形式化されてもよい。例えば、日付及び時間は、共通の基準点のため、要素内の相異なるフィールドとして、又は経過時間単位の数（例えば、秒、ミリ秒など）を表す単一の数として明示的にコード化されてもよい。一部の実施形態において、最近の修正要素は、他の実施形態において、記録１４８内の相異なる要素として含まれてもよいが、ｃｒｅａｔｉｏｎＤａｔｅ要素は、対応するオブジェクト３０の

いかなる局面の最近の修正の日付及び時間を示すように構成される、付加的なフィールドを含んでもよい。

【０１０５】

６４ビットｏｂｊｅｃｔＳｉｚｅ要素は、例えば、バイトで、対応するオブジェクトのサイズの表示を保存するために使用されてもよい。一部の実施形態において、この要素は、他の実施形態において、これらは相異なるフィールドとして保存されてもよいが、オブジェクトデータ及びメタデータの両方のサイズを反映してもよい。３２ビットｃｒｃ３２要素は、あらゆる適切なチェックサムアルゴリズムに従い、オブジェクトデータ及び／又はメタデータのための計算された周期的冗長検査（ＣＲＣ）チェックサムの表示を保存するために使用されてもよい。例えば、チェックサムは、変造、又は改ざんに対するデータ整合性を立証するために含まれてもよい。他の実施形態において、オブジェクトデータ及び／又はメタデータから計算されたあらゆる適切な種類の、ハッシュ、又は署名は、ＣＲＣチェックサムに加えて、又はそれに代わって使用されてもよい。

【０１０６】

８ビットｎｕｍＬｏｃａｔｏｒｓ要素は、複製［］配列内の記録１４８内に含まれるロケータの数の表示を保存するために使用されてもよい。この配列内で、各ロケータは、６４ビットオブジェクトインデックス値のみならず、６４ビットノードＩＤ要素として保存され、ビットストアノード１６０の構造の論考で上記に記載されるように作成されてもよい。一部の実施形態において、ロケータは、複製［］配置内に単一要素として保存されてもよい。

【０１０７】

一実施形態において、キーマップインスタンス１４０は、提供されるＡＰＩをサポートするために必要なこれらの機能を実行するのみならず、コーディネータ１２０などの、キーマップクライアントに対してキーマップＡＰＩを提供するように設定されてもよい。例えば、コントローラ１２０は、キーマップインスタンス１４０によって管理されるエントリ１４４と関連する記録１４８上で、保存、取り出し、削除、又は他の操作を行うために、ＡＰＩを使用するように設定されてもよい。上記に記載されるようにノード１６０によってサポートされてもよいオブジェクトインスタンス上の操作と類似して、一実施形態において、キーマップＡＰＩは、キーマップエントリ１４４上の格納、取得、及び削除操作をサポートしてもよい。このような一実施形態において、キーマップ保存操作、又はキーマップ書き込み操作として総称的に言及されてもよい、キーマップエントリ格納操作は、キーマップエントリ１４４内に保存されるべきキー１４６及び記録１４８を特定してもよい。一実施形態において、エントリ１４４が既に存在するキー１４６を特定する格納操作は、存在するエントリ１４４に関連する記録１４８を、格納操作の引数、又はパラメータとして特定される記録と置き換えてもよい。所定のキーマップインスタンス１４０が完了次第、キーマップ格納操作は、例えば、操作が成功、又は失敗したか、任意の種類の障害が生じたか（もしある場合）などの、状態表示を要求するクライアントに返してもよい。一部の実施形態において、キーマップ格納操作が存在するエントリ１４４の再配置の結果になった場合、キーマップインスタンス１４０は、要求するクライアントに対するエントリ１４４の前回値を返すように設定されてもよい。

【０１０８】

総称的にキーマップ読み込み、又は取り出し操作といわれることもあるキーマップエントリ取得操作は、一実施形態において、パラメータとしてキーを特定してもよい。完了次第、キーマップ取得操作は、当該のエントリが存在する場合、要求されるキーに関連するキーマップエントリ１４４の記録１４８を要求するクライアントに返してもよい。対応するエントリ１４４が存在しない場合は、その要旨が要求するクライアントに返されてもよい。

【０１０９】

一実施形態において、キーマップエントリ削除操作は、要求するクライアントが、エントリに記録を書き込むことを特定する必要がない場合を除き、格納操作と同様に行われる

ように設定されてもよい。所定のキーマップインスタンス１４０が完了次第、キーマップ削除操作は、キーマップ格納操作と同様の状態表示を要求するクライアントに返してもよい。格納操作のように、一部の実施形態において、キーマップインスタンス１４０は、要求するクライアントに対する削除されたエントリ１４４の前回値を返すように設定されてもよい。

【０１１０】

キーマップＡＰＩは、様々な実施形態において、他の種類の操作をサポートしてもよい。例えば、キーマップＡＰＩは、キーマップエントリの管理において、キーマップクライアントを援助する操作をサポートしてもよい。一実施形態において、キーマップＡＰＩは、要求するクライアントによって特定されるいくつかの基準と一致する、キー１４６を有するこれらのエントリ１４４を識別するように設定されてもよいリスト操作を支持してもよい。例えば、リスト操作は、クライアントが、操作に対するパラメータとしての文字列、又はパターンを特定できてもよい。所定のキーマップインスタンス１４０が完了次第、リスト操作は、要求するクライアントに、特定された文字列、又はパターンを満たすこれらのキー１４６のリストを返してもよい。一実施形態において、キー１４６は、キー１４６の適切なプレフィックスである場合にのみ、所定の文字列を満たしてもよい（例えば、文字列のすべての文字について、文字列のＮ番目文字がキーのＮ番目文字と一致する）。他の実施形態において、キー１４６は、文字列が、キー１４６内のいかなる場所において見つかる場合、所定の文字列を満たしてもよい。

【０１１１】

リスト操作は、一部の実施形態において、他のパラメータをサポートしてもよい。例えば、リスト操作は、要求するクライアントが、返されるべき一致数の限度を特定出来てもよい。さらに、要求するクライアントは、例えば、検索すべきキー１４６が当てはまるオープンエンド、又はクローズエンドの辞書式範囲内で、検索するキー１４６に対する制約を特定してもよい。一部の実施形態において、キーマップインスタンス１４０は、リスト操作基準を満たすキー１４６のみならず、記録１４８を返すように設定されてもよい。また、一部の実施形態において、キーマップＡＰＩは、リスト操作として、同一の種類のパラメータ及び実行動作をサポートしてもよい、カウント操作をサポートしてもよい。しかしながら、要求するクライアントによって提供される基準を満たすこれらのキー１４６及び／又は記録１４８を返す代わりに、カウント操作は、これらの基準を満たすキーの数を返してもよい（例えば、対応するリスト操作によって返されるであろうキーの数）。またキーマップＡＰＩは、上記に説明されない他の操作もサポートしてもよいことに留意されたい。

【０１１２】

一部の状況において、異なるキーマップクライアントは、同一のキーマップエントリ１４４を修正することを求める場合がある。例えば、様々なクライアント、又はシステムによる操作に応じ、２つの異なるコーディネータ１２０は、所定の記録１４８の内容を同時に変更することを試みてよく（例えば、複製のロケータを追加、削除、又は修正するために）、又はもう一方が対応するエントリ１４４を削除することを試みる一方、１つが記録１４８を修正することを試みてもよい。所定のキーマップエントリ１４４への同時要求を解決するための一貫した方法を提供するために、一実施形態において、キーマップＡＰＩは、キーマップ状態を更新、又は修正する（例えば、キーマップ格納及び削除操作）少なくともこれらのキーマップ操作が、キーマップ操作に対するパラメータとしての配列番号を提供することを必要としてもよい。キーマップインスタンス１４０は、配列番号を比較すること（例えば、数的に、又は辞書編集的に）、及び比較に基づく操作の１つを継続的に選定することによって、エントリ１４４に対する相反する更新を解決するように設定されてもよい。一部の実施形態において、以下にさらに詳しく説明するように、同期化の回復のために、提供される配列番号は、修正された記録１４８と共に、修正されたキーマップエントリ１４４に保存されてもよい。

【０１１３】

例えば、キーマップクライアントは、タイムスタンプに基づいて配列番号を作成してもよい。一実施形態において、当該のタイムスタンプは、以下のように形式化された６４ビット数を含んでもよい。タイムスタンプのビット６３は、ゼロに設定さてもよい（例えば、タイムスタンプが署名された、又は署名されない番号であるかどうかの混乱を避けるため）。ビット６２：３２は、参照時から経過した秒の数を含んでもよい（例えば、１９７０年１月１日午前零時、グリニッジ標準時、Ｕｎｉｘ及びＬｉｎｕｘの多くのバージョンによって用いられる参照時間）。ビット３１：２２は、最終秒から経過したミリ秒の数を含んでもよい。ビット２１：０は、実質的に無作為に作成されたビットを包含してもよい。他の実施形態において、タイムスタンプは、フィールドの異なる幅、又は種類に基づいて作成されてもよい。代わりに、キーマップクライアントは、配列番号を作成するための、全く異なる基準を用いてもよい。提供される配列番号の解像度が高ければ、同一のキーマップエントリ１４４に対する異なるキーマップクライアントによって提供される、異なる配列番号間のコリジョンの可能性は低い場合がある。しかしながら、コリジョンが発生する場合、キーマップインスタンス１４０は、あらゆる適切な、一貫した技術を使用してコリジョンを解決するように設定されてもよい。

【０１１４】

多くの実施形態において、マッピングキーにおいて、ロケータに対するキーマップインスタンス１４０の抽象的機能性の動作は、比較的直接的である。例えば、上記に記載されるように、キーマップインスタンス１４０の一実施形態によってサポートされる基礎的な操作の一式は、記録１４８内に含まれるキー１４６とロケータとの間の関係を反映するエントリ１４４を操るように構成される、格納、取得、及び削除操作を含んでもよい。しかしながら、ストレージサービスシステム内のキーマップ機能性の実装は多くの困難をもたらす場合がある。とりわけ、ストレージサービスシステムが、多数のクライアントに代行して、膨大な数のオブジェクト３０をサポートする場合（例えば、ストレージの合計でテラバイト（ＴＢ）、又はペタバイト（ＥＢ）になる数百万、又は数十億、又はそれ以上のオブジェクト３０）、同様に容量を拡大するために、キーマップの実装が必要となる場合がある。しかしながら、単独のコンピュータシステム内のキーマップに含まれる全情報を示す、十分なシステムメモリ供給源を実装することは可能でない、又は経済的に実現可能でない場合がある。さらに、フォルトトレランス及び増加するキーマップクライアント要求のための処理スループットについて、キーマップデータの複数の複製が、ストレージサービスシステム内に分散される方法で展開されてもよい。しかしながら、キーマップデータの複製は、例えば、１つの複製が修正される一方、もう１つがアクセスされる場合、キーマップ同期化及び一貫性問題につながる場合がある。

【０１１５】

キーマップ機能性の拡張性は、キーマップインスタンス１４０内の階層のレベルを導入することによって改善されてもよい。このような階層の一実施形態は、図１１Ａ～Ｄに示す。図１１Ａにおいて、例のキーマップ配置１１００が示される。上記に記載されるように、例えば、図３に関し、一部のストレージサービスシステムの実施形態において、複数のキーマップインスタンス１４０が、例えば、異なるデータセンタ３００又は領域３１０に、システム全体にわたり分散されてもよい。概して、キーマップインスタンスの収集物は、展開といわれることもある。一部の実施形態において、他の実施形態においては、システムは、キーマップの階層の付加的なレベル下に統合される複数のキーマップ展開１１００を含んでもよいが、ストレージサービスシステムは、システム内に支給されるすべてのキーマップインスタンス１４０を含む、単一のキーマップ展開１１００を網羅してもよい。

【０１１６】

図示した実施形態において、展開１１００は、キーマップインスタンス１４０ａ～ｃを含み、例えば、以下にさらに詳しく述べるようにインスタンス同期プロトコルに従い、それぞれは他とキーマップ情報を交換するように構成される。示すように、各キーマップインスタンス１４０は、互いに通信しあうように構成される多くのホスト４００を含む。例

えば、キーマップインスタンス１４０ａは、ホスト４００ａ～ｃを含み、キーマップインスタンス１４０ｂは、ホスト４００ｄ～ｇを含み、キーマップインスタンス１４０ｃは、ホスト４００ｈ～ｊを含む。一般的に言えば、各ホスト４００は、コンピュータシステム及び関連するソフトウェアを含んでもよく、プロセッサ、システムメモリ、ストレージデバイス、ネットワーキングインタフェース、又は他の適切なコンポーネントなどの要素を含んでもよい。例えば、ホスト４００としての役割を果たすように構成されるコンピュータシステム、又はノードの一実施形態は、図２９の説明とともに論考される。

【０１１７】

一般的に、各キーマップインスタンス１４０は、ストレージサービスシステム内に保存されるすべてのオブジェクト３０について、キーマップ階層を指数化し、管理するために使用されるあらゆる他のデータのみならず、キーマップエントリ１４４を含む、キーマップデータの完全な表現を維持するように設定されてもよい。キーマップインスタンス１４０内で、キーマップデータはホスト４００にわたって分散されてもよく、個々のホスト４００はキーマップデータの一部の部分（おそらく冗長）を保存してもよい。図１１Ａに、ただわずかのホスト４００が示されるが、他の実施形態において、各キーマップインスタンス１４０は、あらゆる適切な数のホスト１４０を有してもよいことに留意されたい。例えば、一部の大規模な実装において、数ダース、又はことによると数１００のホスト１４０が、キーマップインスタンス１４０に含まれてもよい。一部の実施形態において、所定のキーマップインスタンス１４０のためのホスト４００は、所定の領域３１０、又はデータセンタ３００内にローカライズされるが、他の実施形態において、このようなホスト４００は、異なる領域３１０、又はデータセンタ３００間に分散されてもよいことを考慮されたい。さらに、ホスト４００が、一部の実施形態においてキーマップ関連機能性のみを実装するように構成されるが、他の実施形態において、ホスト４００は、ストレージサービスシステムの他の要素に関連する機能性を実装してもよい。例えば、一実施形態において、様々なホスト４００の１つは、ビットストアノード１６０として設定されてもよく、したがって、オブジェクトデータのみならず、キーマップデータも保存してもよい。

【０１１８】

図１１Ｂは、さらに詳しくキーマップインスタンス１４０ａの例示的な実施形態を示す。図示した実施形態において、キーマップインスタンス１４０ａ内の各ノード４００ａ～ｃは、それぞれのパーティションインデックス４１０ａ～ｃ及び任意の数のブリック４１５を含む。一般的に言えば、ブリック４１５は、キーマップインスタンス１４０内の中間キーマップデータ構造と対応してもよい。一部の実施形態において、図１２の説明とともに以下にさらに詳しく述べるように、キーマップデータは、ブリック４１５の間でパーティションに分割されてもよく、キーマップインスタンス１４０内のパーティションの複製が、ブリックレベルで起こってもよい。パーティションインデックス４１０は、キーマップ操作の間に１つ以上の特定のブリックの処理の選択を容易にするために、ブリック４１５を指数化方式にするように設定されてもよい。例えば、パーティションインデックス４１０は、ツリー、又は別の適切なデータ構造として設定されてもよい。一実施形態において、キーマップインスタンス１４０内のさらに深い指数レベルのみならず、パーティションインデックス４１０は、次に来る項に詳しく説明する、層別不平衡ツリー、又はトライ構造として言及するデータ向上の特定の種類の一部分として設定されてもよい。図示した実施形態において、キーマップインスタンス１４０は、さらにキーマップコーディネータ４１２を含む。一般的に言えば、キーマップコーディネータ４１２は、さらに詳しく以下に説明する、キーマップアクセス管理、コンテント管理、及び同期化方法、又はプロトコルなどを実装するように設定されてもよい。キーマップコーディネータ４１２は、ホスト４００から区別して説明されるが、一部の実施形態において、１つ以上のホスト４００内の処理、又は各要素として実装される場合があることに留意されたい。一部の実施形態において、パーティションインデックス４１０は、ホスト４００内に別々に実装されるよりはむしろ、キーマップコーディネータ４１２内に実装される場合があることも留意されたい

【０１１９】
図１１Ｃは、ブリック４１５ａ～ｎを含む、ホスト４００ａの例示的な実施形態を示す。示すように、各ブリック４１５ａ～ｎは、任意の数のブロック４２５のみならず、それぞれのブロックインデックス４２０ａ～ｎを含む。一般的に言えば、ブロック４２５は、キーマップインスタンス１４０内で、ブリック４１５と類似するが、抽象概念のブリックレベルに対して下位である、中間キーマップデータ構造と対応してもよい。パーティションインデックス４１０と類似して、ブロックインデックス４２０は、ブリック４１５内のブロック４２５の指数化のために構成されるあらゆる適切なデータ構造であってもよい。例えば、ブロックインデックス４２０は、一実施形態において、層別不平衡ツリーの部分として設定されてもよい。

【０１２０】
図１１Ｄに示すように、一実施形態において、ブロック４２５は、選択のためにインデックスエントリ１４４を構成するエントリインデックス４３０のみならず、個々のキーマップエントリ１４４ａ～ｎの任意の数を含むように設定されてもよい。前に説明したように、各エントリ１４４ａ～ｎは、それぞれの記録１４８ａ～ｎのみならず、それぞれのキー１４６ａ～ｎの表示を含んでもよい。

【０１２１】
図１１Ａ～Ｄに示す実施形態のキーマップインスタンス１４０とキーマップエントリ１４４との間の階層層間における関係は、図１２に要約する。複数のキーマップインスタンス１４０を含む抽象概念の展開レベルにおいて、特定のキーマップインスタンス１４０は、抽象概念のインスタンスレベルでパーティションインデックス４１０を参照してもよい。参照されるパーティションインデックス４１０は、特定のエントリ１４４と対応する１つのブリック、又は複数のブリック４１５を識別してもよい。例えば、図示した実施形態において、すべてのキーマップエントリは、相異なるブリック４１５と対応する３つの相異なるパーティションによって複製されてもよい。入れ替わりに、所定のブリックは、ブロックインデックス４２０を介し、特定のブロック４２５（図１２にイは示されない）を参照してもよく、参照されるブロックは、エントリインデックス４３０を介して特定のエントリ１４４を参照してもよい。キーマップが、図１２に示されるように階層の実装を使用して実装されるが、他の実装も可能であることに留意されたい。大まかに言えば、キーマップインスタンス１４０は、記録１４８を有するキー１４４と関連するあらゆる適切な技術を使用して実装されてもよい。例えば、一実施形態において、キーマップインスタンス１４０は、従来のデータ系、又は他の種類の構造化指数を使用して実装されてもよい。

【０１２２】
図１２の実施形態における階層層のいくつかは、他の層が拡張性を提供する様に設定されてもよいが、冗長性を提供するように設定されてもよい（例えば、パーティションレベルでブリック４１５の複製のみならず、展開レベル内でキーマップインスタンス１４０の複製）。例えば、複数の相異なるレベルにわたる指数化の分散は（例えば、パーティションインデックス４１０、ブロックインデックス４２０、及びエントリインデックス４３０）、キーマップ展開内で指数化されるエントリの数が増加する場合、インデックスの各部分が管理可能な方法で増えることを可能にすることで、データ構造の拡大を容易にしてもよい。他の実施形態において、異なる冗長、及び非冗長レベルの組み合せのみならず、さらなる、又はより少ない階層が用いられてもよいことに留意されたい。

【０１２３】
オブジェクト３０のように、キーマップ階層の層内の複製の使用は、個々の複製の損失に対する敏感性を減少すること初期設定トレランスを改善してもよい。しかしながら、修正が発生した場合、キーマップデータの複製を同期化する試みがなされない場合、キーマップの正しい（例えば、最も現在の）状態はあいまいになる場合があり、代わりに予測不可能な、又は誤ったシステム操作につながる場合がある。一部の実施形態において、キーマップデータの複製された部分は、更新が、各複製に関して永久に検証可能な形で完成されるまで、キーマップクライアントに完了が報告されなくてもよい、アトミック、又はト

ランザクションセマンティクスを使用する厳密に同期の方法で更新されてもよい。アトミック更新セマンティクスが、キーマップデータを非一貫性の状態に更新する可能性を最小限にする、またさらには排除してもよいが、アトミック更新の操作は、かなりの規模の分散された環境で、大幅に機能性が低下する場合がある。例えば、キーマップデータの複製が広範囲に分散された場合、更新操作を完了するために必要とされる全体的な時間を決定付ける最も低速の複製では、クライアントから見た複製アクセスの遅延は大幅に異なる場合がある。さらに、１つの複製が失敗した場合、厳密なアトミック更新セマンティクスは、障害が訂正されるまでクライアントを引き止めることを生じる場合があり、クライアントに対する受け入れ難い遅延につながる場合がある。

【０１２４】

アトミックプロトコルよりも、より良いクライアント操作を提供し得る他の種類の同期プロトコルは、キーマップ階層内に用いられてもよい。一部の実施形態において、特定のキーマップインスタンス１４０内の複製に関して一部の種の同期プロトコルが採用されてもよい（例えば、図１２に示すような、特定のレベルにおける複製）、ハイブリッド同期の方法が実装されてもよいが、別の種類のプロトコルが、キーマップ展開内において異なるキーマップインスタンス１４０を同期化するために採用されてもよい。このようなハイブリッド方法は、キーマップ階層内の異なるレベルの複製の使用動学に対して、同期の諸経費を、さらに明確に調整させることを可能にしてもよい。

【０１２５】

例えば、キーマップデータアクセスは、特定のエントリ１４４に対して反復される要求は、別のキーマップインスタンス１４０に対するよりはむしろ、特定のキーマップインスタンス１４０（例えば、地理的、ネットワークトポロジー、又は別の適切な基準において、要求するクライアントに最も近いインスタンス）に向けられる、参照の局所性を示してもよい。すなわち、所定のキーマップインスタンス１４０内のキーマップデータの複製が、異なるキーマップインスタンス１４０における対応するキーマップデータよりはむしろ、所定のクライアントによってアクセスされてもよいということになる場合がある。同様に、一部の実施形態において、所定のキーマップインスタンス１４０内の複製は、相異なるキーマップインスタンス１４０を同期するために使用されるプロトコルよりもさらに迅速に収束するように（例えば、複製間の変更を広める）設定されてもよいプロトコルを使用して、同期化されてもよい。

【０１２６】

一実施形態において、所定のキーマップインスタンス１４０内のキーマップデータ複製の同期化は、適切なバージョンの定数プロトコルを使用して行われてもよい。一般的に言えば、定数プロトコルに従い操作されるキーマップデータの複製の更新、又は修正（キーマップエントリ格納及び削除操作を含む）は、少なくとも定数の数の複製に関して修正が永久的に（例えば、完全に、及び持続的に）行われた時に、要求するクライアントに関して完了すると見なされてもよい。同様に、定数プロトコルに従い行われるキーマップエントリ取得操作は、同一のデータが、少なくとも定数の数の複製から読み込まれた時に完了すると見なされてもよい。一部の実施形態において、定数の数は、存在する複製の数の単純過半数として定義されてもよいが、他の実施形態において、任意の程度の圧倒的多数が用いられてもよい。定数プロトコル操作は、定数要求が満たされない場合は完了しない場合があることに留意されたい。しかしながら、複製の定数の数が複製の合計数よりも少ない場合、所定の定数プロトコル操作が失敗する可能性は、定数よりも複製間での一致を事実上必要とするアトミックプロトコル操作よりも低い場合がある。本願に説明されるもの以外の定数プロトコルは、キーマップインスタンス１４０によって採用されてもよい。例えば、Ｐａｘｏｓなどの複数相コミットプロトコル、又は２相コミットは、定数種キーマップのセマンティクスに用いられる場合があることに留意されたい。

【０１２７】

定数プロトコルに従った読み込み及び更新操作の通常の操作手順において、例えば、通信障害、又は複製の根本をなす供給源の障害により、各複製に更新が伝えることに失敗す

る可能性がある。一実施形態において、複製間における不一致は、読み込み操作の間に検出され、修復されてもよい。とりわけ、キーマップエントリ取得操作の間に、特定のエントリ１４４の異なる複製間で異なる値が検出された場合、キーマップ格納操作は、違いを調整するために作成されてもよい。一実施形態において、格納操作の基準として使用されるエントリ１４４は、異なる値読み込み間におけるタイムスタンプと関連する最新の（例えば、数的に、又は辞書編集的に最も高い）エントリであってもよい。したがって、複製間の相違は、相違の修復のための相異なる過程、又は操作を必要とすることなく、例えば、キーマップエントリ取得操作が処理される場合、「オンザフライ」で解決されてもよい。

【０１２８】

定数プロトコルの実装を構成するキーマップインスタンス１４０の実施形態に関するキーマップエントリ格納、取得、削除、及びリスト操作の例示的な実施形態の操作は、図１３から図１４に示される。様々な実施形態において、これらの方法は、例えば、キーマップインスタンス１４０内に含まれる１つ以上のホスト４００内に設定されてもよい、又は図１１Ｂに示すキーマップコーディネータ４１２のような、キーマップインスタンス１４０内の別個の過程、又はシステムとして構成される、キーマップコーディネータ処理内に実装されてもよい。最初から図１３までに言及して、キーマップエントリ格納操作は、該操作がコーディネータ１２０又は他のキーマップクライアントからキーマップインスタンス１４０で受信される時、ブロック１３００で始まってもよい。例えば、特定のオブジェクト３０の対応するオブジェクトインスタンスを特定のビットストアノード１６０に保存することに応じ、コーディネータ１２０は、保存されたオブジェクトインスタンスのロケータを反映するためのオブジェクト３０のエントリ１４４を更新するため、キーマップエントリ格納操作を作成してもよい。

【０１２９】

キーマップインスタンス１４０の階層は、キーマップエントリ格納操作と対応する複製を識別するために誘導さてもよい（ブロック１３０２）。例えば、図１２の実施形態について、パーティションインデックス４１０は、どのブリック４１５が、重要なオブジェクト３０と対応するエントリ１４４を複製するかを決定するように設定されてもよい。続いて、個々の格納操作が、識別された複製に指示されてもよい（ブロック１３０４）。各格納操作について、キーマップインスタンス１４０の残存する階層は、対応するエントリ１４４へのアクセス及び修正を誘導されてもよい（ブロック１３０６）。例えば、所定のブリック４１５内でブロックインデックス４２０及びエントリインデックス４３０は、特定されたエントリ１４４にアクセスするために横断されてもよい。エントリ１４４の所定の複製が無事に書き込まれると、対応する格納操作が成功を表示する（ブロック１３０８）。エントリ１４４の目標とするそれぞれの複製の個々の格納操作は、同時に行われてもよい。同様に、ブロック１３０６～１３０８の複数のインスタンスは平行に示される。

【０１３０】

個々の複製格納操作の成功表示は、複製の定数の数が無事に更新されたかを決定するために監視されてもよい（ブロック１３１０）。例えば、３つの複製を含む実施形態において、キーマップエントリ格納操作の完了のための複製の定数の数は、２であってよい。複製の定数の数が無事に更新された場合、要求されたキーマップエントリ格納操作が完了した表示は、要求するクライアントに返されてよい（ブロック１３１２）。そうでない場合、監視が続いてもよい。一部の実施形態において、キーマップエントリ格納操作が、処理の開始後特定の期間内に完了しない場合、操作は終了され、エラー表示が要求するクライアントに返される、タイムアウトが強制されてもよい。他の実施形態において、キーマップエントリ格納操作は、それが完了するまで永久に保留を維持してもよい。

【０１３１】

一実施形態において、キーマップエントリ削除操作は、格納操作の特別な場合として実装されてもよい。このような実施形態においてキーエントリ１４４は、削除センチネル又はフラッグフィールドを構成する付加的なフィールドを含んでもよく、削除操作は、削除

フィールドをアサートされた状態（例えば、「１」などの特定の値をフィールドに設定することによって）に設定するように構成される、格納操作として実行されてもよい。アサートされた削除フィールドを有するこれらのエントリ１４４は、将来のキーマップ操作中に無視されてもよい。このような一部の実施形態において、アサートされた削除フィールドを有するこれらのエントリ１４４を消去するためにキーマップインスタンス１４４を通じて独立して反復されるように設定されてもよい。他の実施形態において、当該エントリ１４４は、過去のキーマップ動作のログとして永久に保持されてもよい。

【０１３２】

キーマップエントリ格納操作の方法の一実施形態は、図１４に示す。操作は、取得操作が、コーディネータ１２０又は他のキーマップクライアントからキーマップインスタンス１４０で受信される時、ブロック１４００で始まってもよい。例えば、特定のキーと対応するオブジェクトデータに対するストレージクライアント５０からの要求に応じて、ノードピッカー１３０又はコーディネータ１２０は、特定のキーと対応するロケータを取得するためにキーマップエントリ取得操作を作成し、それによってビットストアノード１６０が、前項に説明されるようにオブジェクトデータを取り出すためにアクセスされてもよい。

【０１３３】

キーマップエントリ格納操作のように、キーマップインスタンス１４０の階層は、キーマップエントリ取得操作と対応する複製を識別するために誘導さてもよい（ブロック１４０２）。続いて、個々の取得操作が、識別された複製に指示されてもよい（ブロック１４０４）。各取得操作について、キーマップインスタンス１４０の残存する階層は、対応するエントリ１４４へのアクセス及び取り出しを誘導されてもよい（ブロック１４０６）。エントリ１４４の所定の複製が無事に取り出されると、対応する取得操作は成功を表示してもよい（ブロック１４０８）。上記、及び図１３に示す個々の格納操作のように、個々のエントリ１４４のそれぞれの複製を目標とする取得操作は、同時に行われてもよく、ブロック１４０６～１４０８は、同様に平行で示される。

【０１３４】

個々の複製取得操作の成功表示は、複製の定数の数が無事に読み込まれたかを決定するために監視されてもよい（ブロック１４１０）。そうでない場合、付加的な複製が読み込まれるまで監視が続いてもよい。キーマップエントリ格納操作について上記に記載されるように、一部の実施形態において、キーマップエントリ取得操作は、定数の数の複製が無事に読み込まれるまで永久に待機してもよい。他の実施形態において、キーマップエントリ取得操作は、一定期間の後、タイムアウトになってもよく、エラー表示後及び／又はその時点で利用可能な最良のデータ（例えば、最も最近のタイムスタンプを有する複製データ）が、要求するクライアントに返されてもよい。

【０１３５】

複製の定数の数が無事に読み込まれた場合、取り出された複製の内容が異なるかどうか決定されてもよい（ブロック１４１２）。例えば、要求されたエントリ１４４の各複製の全部は、他の取り出された複製のそれぞれと比較されるか、又はエントリ１４４のあるフィールドのみ（例えば、記録１４８のあるフィールド）比較されてもよい。該比較に使用された基準に従い、取り出された複製間に相違がない場合、取り出されたデータは、キーマップエントリ取得操作が完了したことの表示とともに、要求するクライアントに返されてもよい（ブロック１４１４）。

【０１３６】

複製間の相違が存在する場合、選択基準に従い、複製の１つが選択されてもよい（ブロック１４１６）。例えば、該基準は、最も高いタイムスタンプ値を有する複製が選択される、各複製のタイムスタンプ値を考慮することを含んでもよい。キーマップエントリ格納操作は、選択された複製のデータを使用して開始されてもよい（ブロック１４１８）。例えば、格納操作は、上記に記載されるように図１３に従い行われてもよい。格納操作の結果として、初めに要求されたエントリ１４４の複製の定数の数は、選択された複製の内容

で書き込まれてもよく、将来の取得操作が、複製間の不一致に遭遇するのをを減少する。格納操作に続き、選択された複製のデータが、キーマップ取得操作が完了した表示と共に、要求するクライアントに返されてよい（ブロック１４１４）。一部の実施形態において、複製間で検出された不一致の場合の取得操作の完了は、不一致を解決するための開始された格納操作の完了を条件としてもよいが、他の実施形態において、取得操作は、カットして生じる格納操作が完了したかどうかに関係なく、完了したことを要求するクライアントに表示する

【０１３７】

上記に記載されるように、一部の実施形態において、キーマップＡＰＩは、検索パターンなど、一部の基準を満たすキーマップエントリ１４４のこれらのキー１４６を示すように構成される、キーマップエントリリスト又はカウント操作をサポートしてもよい。一実施形態において、リスト及び／又はカウント操作は、キーマップエントリ取得操作の特別な場合として実装されてもよく、所定のリスト、又はカウント操作の基準を満たす各エントリ１４４について、対応するキーマップエントリ取得操作が行われてもよい。しかしながら、定数プロトコルに従い、複数の複製からエントリデータ（例えば、記録１４８）を実際に取り出すことの付加的な諸経費は、キーマップエントリリスト、又はカウント操作に不必要である場合がある。したがって、一部の実施形態において、定数プロトコルに関係するキーマップエントリ取得操作のこれらのステップは、キーマップエントリリスト、又はカウント操作から省略されてもよい。例えば、所定のエントリのすべての複製を識別し、ブロック１４０２～１４０４の各複製のための個々の取得操作を作成するよりはむしろ、リスト、又はカウント操作について、単一の複製は（例えばブリック４１５）任意の選択されてもよく、その対応する階層が、リスト、又はカウント操作基準を満たす各エントリ１４４を識別するために誘導されてもよい。該基準を満たす結果として得られるエントリ１４４について、結果として得られるエントリ１４４の対応するキー１４６、又はカウントは、図１４の定数関係の処理部分を回避して、要求するクライアントに返されてもよい（例えば、ブロック１４１０～１４１８）。

【０１３８】

一部の実施形態において、キーマップインスタンス１４０は、インデックスエントリ１４４に対して使用された様々なデータ構造に加え、キャッシュを実装してもよい。例えば、キャッシュは、頻繁に使用されるエントリ１４４のキーに命令されたキーマップ操作が、対応するエントリ１４４に直接アクセスするために、インデックスデータ構造の誘導を回避することを可能にしてもよく、キーマップエントリ取得操作の性能を改善してもよい。さらに、キャッシュは、普及した、頻繁にアクセスされるキーに関連するホスト４００を、キーマップ要求トラフィックによって過負荷になることから防ぐ助けとなってもよい。例えば、キーマップキャッシュがホスト４００の間に分散される一実施形態において、キーのコピーは、キーのためのインデックスデータ構造を維持するホスト４００よりも異なるホストでキャッシュされてもよい。ホスト４００の間でキャッシュするキーのこのような分散に関わらず、キー処理作業負荷は、ホスト４００の間でさらに均一に分けられてよい。

【０１３９】

一実施形態において、キーマップキャッシュは、キーそのものよりも、キー１４８のハッシュによって保存及びインデックス化されるように設定されてもよい。データハッシュは、不均衡インデックスデータ構造の論考と共にさらに以下に詳しく説明するように、キーマップキャッシュ内で管理することがさらに容易であるような、キー１４８などの、確定された長さのデータ構造において、様々な長さのデータを表すための効果的な技術を構成してもよい。さらに、様々なハッシュアルゴリズムは、当初は均一に分散されていない場合があるデータ（例えば、共通するデータの多量の部分を有するキー１４８の一式）のための均一に分散されたハッシュ値を作成してもよく、ホスト４００間におけるキーマップキャッシュデータの一定の分散を容易にしてもよい。一部の実施形態において、エントリ１４４の内容は、対応するクー１４８のハッシュされた値と共にキーマップキャッシュ

に保存されてもよい。他の実施形態において、エントリ１４４それ自体の内容よりはむしろ、エントリ１４４のためのポインタ、又は他の参照情報が保存されてもよい。

【０１４０】

一般的に言えば、キーマップキャッシュを含むキーマップ実施形態において、キーマップエントリ格納及び取得操作は、上記説明に対するわずかな修正をともなって行われてもよい。一実施形態において、キーマップエントリ取得操作は、取得操作が、キャッシュのデータ常駐から情報提供されることができるかどうかを決定するために、最初にキャッシュを参考にしてもよい。所得操作は、読み込みのための定数プロトコルに移る前に、キャッシュからの応答のために一定量の時間待機してもよい。キャッシュが、定数プロトコル読み込みが開始された後に値を返した場合、キャッシュからの値読み込みが処理され、対応するエントリ１４４が返され、定数プロトコル読み込みが終了されてもよい。キャッシュから値が返されない場合、定数プロトコル読み込み操作からのエントリ１４４読み込み、又は当該のエントリ１４４に対するポインタは、対応するキー情報と共にキーマップキャッシュにインストールされてもよい。

【０１４１】

一般的に言えば、キャッシュを含むキーマップ実施形態におけるエントリ格納操作は、同一のキャッシュエントリを修正するために、複数の格納操作を同時に試みることを妨げるためにロックされ、又は他の一貫性プロトコルが採用されることを除き、上記に記載されるように実質的に行われてもよい。一実施形態において、キーマップエントリ操作は、書き込みのための定数プロトコルを開始する前に、キー１４８と対応するキャッシュエントリをロックすることを試みるように設定されてもよい。ロック要求が成功したキャッシュからの応答を受け取ると（例えば、エントリに他にロックが存在しないため、又はキャッシュに対応するエントリがないため）、定数プロトコルを続行してもよい。定数プロトコルに従い納操作が完了した後、ロックが解放され、新規エントリデータがキャッシュにインストールされてもよい。

【０１４２】

一部の実施形態において、今しがた説明したように、キーマップエントリ格納及び取得操作のための定数プロトコルは、キーマップエントリ状態を更新するための強一致性モデルを実装してもよい。すなわち、定数プロトコルは、特定のキーに対する格納操作が、完了するときにクライアントに認識されるとそれに続く取得操作が、取得操作が続行された時点において、各複製が更新されてない場合でも、最も最近格納されたデータを返すことを保証してもよい。

【０１４３】

格納及び削除操作などの、キーマップ操作は、特定のキーマップインスタンス１４０にたいして命令され、その特定のキーマップインスタンス１４０内のエントリ１４４の状態は、時間と共に変わってもよい。したがって、それらを調整する試みがないと、展開内の異なるキーマップインスタンス１４０は、時間と共に相違、又は不一致になる傾向がある場合がある。たった１つのストレージサービスクライアント５０が、所定のオブジェクト３０を参照する場合、またそれを同一のキーマップインスタンス１４０を介して行う場合、当該の相違は実用上の影響を有さない場合がある。しかしながら、複数のストレージサービスクライアント５０が、異なるキーマップインスタンス１４０を介して同一のキーを参照する場合、当該の不一致は、クライアント５０が、同一の時点において、異なるキーマップ状態及び／又はオブジェクトデータの異なるバージョンを観察する結果となる場合がある。

【０１４４】

前に説明したように、アトミックプロトコル、又は定数プロトコルのような強一致性プロトコルは、クライアントが複製の不一致を観察することを効果的に防ぐために、又は当該の不一致が起こることを完全に防ぐために、複製を更新する時に用いられることがある。しかしながら、異なる複製のアクセス待ち時間が異なる分散された状況において、時々多量の強一致性プロトコルが非常に高い操作コストを有してもよい。例えば、アトミック

プロトコル、又は定数プロトコルについて、操作完了に必要とされる時間は、それぞれ、すべての複製の最も遅いもの、又は複製の定数の数の最も遅いものに関して、操作を完了するために必要とされる時間の機能であってもよい。さらに、強一致性プロトコルがないことにおいて、クライアントに対して複製の不一致が可視になることの見込みは（例えば、ストレージサービスクライアント５０が古くなったキーマップ、又はオブジェクトデータを得ることの可能性）、概して、アクセスされた複製がまだ更新を反映しない期間内に、クライアントが複製にアクセスすることの見込み機能であってもよい。

【０１４５】

多くのオブジェクト３０について、この後者の見込みは低くてもよい。例えば、一部の例において、大部分のオブジェクト３０は、特定のキーマップインスタンス１４０を介する単一のクライアント５０によってアクセスされてもよいストレージサービスシステムによって管理され、この場合、不一致は、クライアントの観点からは現実的な価値を無くす場合がある。複数のクライアント５０からアクセスされてもよいオブジェクト５０について、観察できる不一致は、依然として可能性が低い。例えば、２つのキーマップインスタンス１４０は、例えば、１０秒間の期間、特定のキーに関して不一致である。しかしながら、不一致の期間に関してアクセスが行われない場合（例えば、対応するオブジェクト３０のアクセス間の持続時間が不一致の期間よりも長い場合）、又は行われるアクセスがさらに最近更新されたキーマップインスタンス１４０に命令された場合（例えば、キーの状態を最後に更新したクライアント５０が、同一のキーマップインスタンス１４０を介するキーを参照するための次である場合）、不一致はクライアント５０に目立った影響を有さない場合がある。したがって、一部の実施形態において、キーマップインスタンス１４０は、キーマップインスタンス１４０を一貫した状態に収束するよう努める緩やかな同期プロトコルを採用してもよいが、いかなる時において、キーマップインスタンス１４０の間にある程度の不一致を与える場合がある。この様な同期プロトコルは、厳密な同期化が必要でないような、大部分のクライアント５０に対して、よりよい全体的な機能を提供してもよい。一部の実施形態において、共有されるオブジェクト３０のためのキーマップデータの厳密なアクセス同期化を必要とするクライアント５０は、すべてのクライアント５０が厳密な同期化の義務を課すことを要求することなく、クライアントの間で付加的なプロトコルを実装してもよい。例えば、特定のオブジェクト３０の１組へのアクセスを共有するクライアント５０の１組は、キーマップデータへのアクセスを調整するセマフォ、又は他の分散されたロック技術を採用してもよい。

【０１４６】

一部の実施形態において、キーマップインスタンス１４０の間の緩やかな同期プロトコルは、同期化過程の異なる態様を独立して実行する、異なる同期化タスクの組み合せを含んでもよい。図１５Ａ～Ｂは、２つの相異なる同期化タスク、図１５Ａに示す更新伝播タスクと、アンチエントロピー、又は図１５Ｂに示す設定調整タスクを含む、緩やかな同期プロトコルの操作の方法の一実施形態を示す。図１５Ａに最初に言及して、１つのキーマップインスタンス１４０に更新するブロック１５００で始まる操作が検出されてもよい。例えば、キーマップインスタンス１４０は、上記に記載されるように、定数プロトコルに従い、キーマップエントリ格納、又は削除操作を受信し、完了してもよい。

【０１４７】

キーマップ更新を処理したキーマップインスタンス１４０は、ストレージサービスシステム内でプロビジョニングされたそれぞれの他のキーマップインスタンス１４０に対する更新操作を転送してもよい（ブロック１５０４）。例えば、キーマップインスタンス１４０ａがキーマップエントリ格納操作を処理した場合、キーマップインスタンス１４０ｂと１４０ｃに引数、パラメータなどを含む操作を転送してもよい。一実施形態において、転送は照合、又は認識なしで行われてもよい。例えば、キーマップ更新操作を処理したキーマップインスタンスは、転送された操作がその送り先で受信されかたどうかを照合すること、又はそれが受信されていない場合には操作を再送信することを試みること無しで互いのキーマップインスタンスに操作を転送することを一度だけ試みる、「ファイアー・アン

ド・フォーゲット」プロトコルを使用する操作を転送してもよい。この様な転送は、源を発するキーマップインスタンス１４０から複数のキーマップインスタンス１４０への同時送信、源を発するキーマップインスタンス１４０から他のインスタンスへの配列転送、ツリー系方策などの、あらゆる適切な転送方策を使用して生じてもよい。

【０１４８】

転送された操作を受け取るこれらの関連するホスト４００は、更新操作をローカルで実行してもよい（ブロック１５０６）。例えば、ホスト４００ｆが、ホスト４００ａから転送されたキーマップエントリ格納操作を無事に受信した場合、それはあらゆるキーマップクライアントから操作を受信したかのように操作を実行してもよい。ホスト４００ｆで格納操作が無事に完了した場合、その結果として、キーマップインスタンス１４０ａと１４０ｂは、格納操作に関して同期化されてもよい。

【０１４９】

一般的に言えば、ホスト４００間における転送するキーマップ更新操作は、大多数の場合成功する。したがって、このような操作を転送することを伴う諸経費を最小限にすることは、大多数の事例において、キーマップインスタンス１４０間における同期化を達成するために必要とされる時間及び／又は回線容量を削減する。例えば、転送過程から、肯定応答、又は他の種類の照合プロトコル、又はハンドシェイクを消去することは、さらに大きな程度の同期化トラフィックを伴う、キーマップ実装のさらに大きい規模をサポートすることなど、他のユーザのための伝達回線容量を自由にする。多くの例において、キーマップ展開を通じてキーマップ更新を展開するために必要とされる時間は（例えば、所定のキーマップエントリ１４４の複製の不一致の可能性のウィンドウと概して対応してもよい）、ホスト４００と関連する操作を転送するために必要とされる伝達の遅延、ホスト４００が転送された操作を適応する為に必要とされる処理の遅延を制限してもよい。しばしば、この合計時間は、秒の順序、又は秒の分数であってもよい。

【０１５０】

しかしながら、一部の例において、ホスト４００間におけるキーマップ更新操作の転送は失敗する場合がある。例えば、伝達リンクの障害は、別のホストから１つのホスト４００へ達し得ないことを伝える場合があり、又は転送された操作の損失、不完全化、又は輸送中に破損することを生じる場合がある。代わりに、送り先ホスト４００は、例えば、一時的なハードウェア又はソフトウェアの問題のため、適切に転送された更新操作を受信し、又は正しく処理することに失敗する場合がある。一実施形態のように、転送されたキーマップ更新操作が、目的とされるホスト４００によって無事に受信され、処理されたことを照合する、又は確かめるために、源を発するホスト４００の部分になにも試みがなされない場合、個々の操作の転送の失敗は、あるエントリ１４４に関するキーマップインスタンス１４０間における不一致の結果になる場合がある。

【０１５１】

同様に、一実施形態において、キーマップインスタンス１４０間における緩やかな同期プロトコルは、上記に記載されるように、及び図１５Ｂに示すように、アンチエントロピー、又は設定調整タスクを含んでもよい。このタスクは、概して、タスクの操作は、異なるキーマップインスタンス１４０間における相違点を減少し、類似点を増加する役割を果たす場合があり、したがって、適切に同期化されたインスタンスに対する更新伝播の無作為な、又は体系的な障害によってもたらされることがある、キーマップインスタンス１４０間における全体的なエントロピーの減少となる、「アンチエントロピー」タスクといわれることもある。図示した実施形態において、操作はブロック１５１０で始まり、開始しているキーマップインスタンス１４０が、特定のパーティションの調整を行う、別のキーマップインスタンス１４０を無作為に選択し、異なるホスト４００における

【０１５２】

開始しているキーマップインスタンス１４０は選択されたキーマップインスタンス１４０を有するインスタンス内のパーティションについての情報を交換してもよい（ブロック１５１２）。例えば、２つのキーマップインスタンス１４０内の特定のホスト４００は、

各インスタンス内で維持されるパーティションインデックス４１０のコピーを交換し、入れ替わりに各インスタンス内のこれらのブリック４１５を識別するように設定されてもよい。

【０１５３】

交換されたパーティション情報に基づき、開始しているキーマップインスタンス１４０は、２つのインスタンスのパーティション間での一致を識別してもよく（ブロック１５１４）、選択されたキーマップインスタンス１４０内の対応するパーティションを有する源を発するキーマップインスタンス１４０内の各パーティションを調整してもよい（ブロック１５１６）。例えば、前に説明したように、所定のキーマップインスタンス１４０内の各パーティションは、多くのブリック４１５に渡って複製されてもよい。一実施形態において、開始しているキーマップインスタンス１４０は、パーティション間の相違を調整するために、パーティション内の特定のブリック４１５が（「リードブリック」といわれることもある）選択されたキーマップインスタンス１４０内の対応するパーティションの対応するブリック、又は「ピア」ブリック４１５と通信することを命令するように設定されてもよい。一実施形態において、２つのブリック４１５の調整は、ブリックが各ブリック４１５に含まれるキーマップエントリ１４４の相違についての情報を交換することを伴い、各キーマップインスタンス１４０内の最も現在の情報を展開することを伴ってもよい。例えば、１つのブリック４１５が、タイムスタンプ情報に基づき、エントリ１４４のバージョンがピアブリック４１５よりもさらに現在のものであることを決定する場合、ブリックはエントリデータからピアブリック４１５に対して通信してもよい。それに続き、ピアブリック４１５は、エントリ１４４のそのコピーを更新するために、キーマップエントリ格納操作（例えば、上記に詳しく説明した定数プロトコルに従い）を行ってもよい。

【０１５４】

２つのキーマップインスタンス１４０間のパーティション調節が完了すると、操作は、調整操作が、別のランダムキーマップインスタンス１４０に関し再び開始されるブロック１５１０から継続されてもよい。様々な実施形態において、各キーマップインスタンス１４０は、予め所定の、又は動的に決定された間隔でこの課程を実行するように設定されてもよい。例えば、調整は、１分に１回の静的割合、又はランダム確率分布、又は他の統計的確率分布に従い決定される間隔で起こってもよい。一部の実施形態において、ある数のキーマップアクセスが起こった後に、又はある個々の１つにアクセスした後に、キーマップエントリの種類、又はグループが検出された後に調整が行われてもよい。

【０１５５】

一般的に言えば、図１５Ａ～Ｂに示す更新伝播及び調整の設定又はアンチエントロピーの方法は、補足的な方法で行われてもよい。大部分の状況下において、更新伝播は、展開内において異なるキーマップインスタンス１４０を満足に同期化してもよい。キーマップの不一致が更新伝播の障害によりもたらされるこれらのインスタンスにおいて、アンチエントロピータスクは、当該の不一致を調整するために概して行われてもよい。一部の実施形態において、アンチエントロピータスクの実行は、２つのキーマップインスタンス１４０がその全体において正確に同期化されることを保証しない場合があることに留意されたい。しかしながら、一実施形態において、アンチエントロピータスクは、その実行が２つのキーマップインスタンス１４０間の不一致の程度を増加しないことを保証するために実装されてもよい。したがって、度重なるアプリケーションの全体に渡って、アンチエントロピータスクは、キーマップインスタンス１４０の収束を容易にしてもよい。アンチエントロピータスクの一実施形態におけるさらなる詳細は、キーマップインスタンス１４０が実装されてもよいデータ構造の特定の実施形態の説明とともに以下に提供される。

【０１５６】

図２に示し、上記で論考したように、一部の実施形態において、ストレージサービスシステムは、他のキーマップインスタンス１４０に加え、レプリケータキーマップインスタンス１９０を含んでもよい。一実施形態において、レプリケータキーマップインスタンス１９０は、上記に記載されるとおり、キーマップインスタンス１４０と基本的にあらゆる

点で等しく構成され、上記で論考したように、プロトコルを使用してキーマップ同期化に関与してもよい。しかしながら、この様な実施形態において、レプリケータインスタンス１９０は、コーディネータ１２０、又は他のキーマップクライアントよりはむしろレプリケータ１８０の役目をするように設定されてもよい。一部の状況において、他のキーマップインスタンス１４０から分離するレプリケータキーマップインスタンス１９０は、概してキーマップ能力を改善してもよい。例えば、レプリケータ１８０は、オブジェクト３０の複製のヘルス状態及び数をチェックするためにキーマップを通じて反復する、相当量のキーマップ要求トラフィックを作成してもよい。ストレージサービスクライアント５０の要求を代行して作成されたキーマップトラフィックと混合されると、レプリケータキーマップトラフィックは、応答時間、又は他のクライアント５０に関するサービスの質の目安にマイナスの影響を与える場合がある。それに反して、戦況のキーマップインスタンス１９０を使用するためにレプリケータ１８０を構成することは、クライアント作成のトラフィックから内部で作成されたキーマップトラフィックを分離させてもよい。加えて、このような分離は、その大部分のクライアントの要求に従い、各種類のキーマップインスタンスを増やすことを実装することをより可能にする。例えば、レプリケータキーマップインスタンス１９０の実装は、いかなる所定のキーマップ操作の遅延を最小にするためよりはむしろ、多数の同時キーマップ操作の処理を容易にするように設定され、キーマップインスタンス１４０は、サービスの質基準の異なる組み合わせのために最適化される。しかしながら、この方法におけるキーマップインスタンスの分離は必要ではなく、一部の実施形態対において、レプリケータ１８０は、専用のレプリケータキーマップインスタンス１９０よりはむしろ、キーマップインスタンス１４０のクライアントであってもよいことに留意されたい。

【０１５７】

一実施形態において、レプリケータキーマップインスタンス１９０は、クライアント５０によるストレージサービスシステム供給源の使用の会計を容易にするように設定されてもよい。具体的に、レプリケータキーマップインスタンス１９０は、対応するオブジェクト３０に対する請求、又は他の金融責任を負うそれぞれのエンティティを示す付加的なデータを有する、キーマップインスタンス１４０によって保存されたエントリ１４４を増やすように構成されてもよい。例えば、図１６に示した実施形態において、レプリケータキーマップエントリ１９４が示される。 entry 194 may function identically to entries 144 with respect to the structure and hierarchy of keymap instances 140レプリケータキーマップインスタンス１９０内において、エントリ１９４は、キーマップインスタンスの構造及び階層に関して、エントリ１４４に対してあらゆる点で等しく機能してもよい。しかしながら、具体的な実施形態において、エントリ１９４は、付加的なフィールド、バケットＩＤ１９６を含む。一般的に言えば、バケットＩＤ１９６は、キー１４６に対応するオブジェクト３０を含むバケット２０の識別子の表示を含んでもよい。当該識別子は、例えば、上記に記載されるようにオブジェクト３０を保存するためのバケット２０を作成するためにクライアント５０からの要求に応じる、ウェブサービスインターフェース１００又はコーディネータ１２０によって定義されてもよい。他の実施形態において、会計情報は、レプリケータキーマップインスタンス１９０のエントリ内だけを反映する必要はないことに留意されたい。例えば、一実施形態において、一部の、又はすべてのキーマップインスタンス１４０のキーマップエントリ１４４は、例えば、記録１４８、又はキー１４６内の付加的なフィールドとして、バケットＩＤ１９６の表示を保存するように設定されてもよい。

【０１５８】

上記に記載されるように、オブジェクト３０とバケット２０との間の関係は、キーマップインスタンス１４０の全体的な操作に対し、透過的であってもよい。しかしながら、もし、この関係が一般的に静的である場合、レプリケータキーマップエントリ１９４を介して明示的に関連するバケット２０及びオブジェクト３０は、クライアント５０の会計及び請求を容易にする場合がある。例えば、各オブジェクト３０と関連するバケット２０のた

めのウェブサービスインターフェース１００を明示的に要求するよりはむしろ、会計過程（レプリケータ１８０、又は別のモジュールに含まれてもよい、又はシステム内の相異なるボジュール内に実装されてもよい）は、バケットＩＤ１９６に従い、レプリケータキーマップエントリ１９４をソートするように設定されてもよい。当該のソートが完了すると、特定のバケットＩＤと関連するすべてのキー１４６は容易に明らかになる。記録１４８内に表示されたように対応するオブジェクト３０のサイズは、バケットＩＤ１９６に関連する全体のストレージリソースの使用を決定するために合計される。さらに、特定のオブジェクト３０と関連するストレージのクラスなど、オブジェクト３０の他の特徴が配慮されてもよい。供給源の使用が、適切な請求モデルに従い通貨に定めされてもよい。

【０１５９】

様々な実施形態において、レプリケータキーマップエントリ１９４は、様々な内部システム管理、又は会計タスクを容易にしてもよいバケット１９６のかわりに、又はそれに加えて他のフィールドを含んでもよい。レプリケータキーマップインスタンス１９０が他のキーマップインスタンス１４０から相異なる実施形態において、当該付加的なフィールドのストレージコストはレプリケータキーマップインスタンス１９０を構成してもよいことに留意されたい。しかしながら、専用のレプリケータキーマップインスタンス１９０を欠く実施形態において、キーマップインスタンス１４０のエントリ１４４は当該付加的なフィールドを含むために増大されてもよい。

【０１６０】

層別不平衡データ構造

前に説明したように、一部の実施形態において、ストレージサービスシステムは、例えば、数十億又はそれ以上の、非常に多いオブジェクト３０の数をサポートするために増やしてもよい。したがって、このような実施形態において、各キーマップインスタンス１４０は、管理するための同様の数のエントリ１４４を有するであろう。一部の実施形態において、キーマップインスタンス１４０は、前項で論考したキーマップエントリリスト、及びカウント操作などの、様々な種類のソート化及び／又はグループ化操作をサポートしてもよい。さらに、一貫したキーマップ操作をサポートするために、各キーマップインスタンス１４０によって管理される多くのキーは、上記に記載されるように他のキーマップインスタンス１４０の間で同期化される必要があってもよい。

【０１６１】

多くの状況において、キーマップインスタンス１４０によって提供されたキーマップ機能性は、全体的なストレージサービスシステムの操作の中心である。例えば、クライアント５０がオブジェクト３０の特定のインスタンスに対してロケータ系のアクセスを行わないことを選ぶ場合、キーマップインスタンス１４０はクライアント５０によって操作される各キー系のオブジェクトアクセスを仲介してもよい。したがって、クライアント５０によって見られるストレージサービスシステムの操作は、アクセスするキーマップインスタンス１４０及びキーマップエントリ１４４の処理の効率及び速度に直接左右される場合がある。言い換えると、キーマップインスタンス１４０の性能は、図１２の実施形態の、パーティションインデックス４１０、ブロックインデックス４２０、エントリインデックス４３０を実装するために使用されるデータ構造などの、エントリ１４４をインデックス化し、体系化するデータ構造に直接左右される。

【０１６２】

大規模のキーマップ実装において、ソート化及び同期操作をサポートするためのインデックスデータ構造を設計することは、相当量の課題をもたらす場合がある。例えば、データ系など、大量のデータのインデックス化を必要とする従来のアプリケーションは、Ｂツリー、又は他の種類の平衡ツリーなどの従来の平衡データ構造をしばしば用いる。一般的に言えば、キーマップエントリ１４４などの、所定の数量のデータアイテムをインデックス化するために使用する場合、平衡データ構造アルゴリズムは、管理するべきアイテムの数量に従い、平衡データ構造にわたって、データアイテムを分散する試みをする。例えば、所定の１００００のキーマップエントリ１４４をインデックス化するためには、平衡デ

ータ構造アルゴリズムは、エントリが１グループおよそ１０００のエントリの１０のグループに分割されるような、エントリ１４４の間の区切り点を選ぶ試みをしてもよい。平衡データ構造アルゴリズムは、例えば、およそ１０００のエントリの各グループを、それぞれおよそ２００のエントリの５つの下位グループに再分割する、各グループ内に平衡した階層のさらなるレベルを作ってもよい。データアイテムが平衡データ構造に加えられ、データ構造から削除されると、データ構造内のグループ及び／又は下位グループは不平衡となる場合がある。したがって、従来の平衡データ構造アルゴリズムは、グループ間でデータアイテムを再配置することによって、付加的なグループを作ること、及び／又は階層の付加的なレベルを作ることによって、データ構造を再平衡する。このような再平衡は、データアイテムが加えられ、又は削除された場合、「オンザフライ」が行われる、又はある数のデータアイテム修正が行われたあと、又は最後の再平衡から一定期間が経過した後に「オンザフライ」が起こる場合がある。

【０１６３】

平衡の取れた方法においてデータを分離することの長所によって、平衡データ構造は、予測可能な、データ構造内でのいかなる所定のデータアイテムのおよそ一定のアクセス遅延をもたらしてもよく、多数のデータアイテムをインデックス化することが必要である、大規模の実装において望ましい場合がある。しかしながら、例えば、上記に記載されるように緩やかな同期化モデルを使用することは、平衡データ構造の分散されたインスタンスを効果的に調整、又は同期化することは特に困難である場合がある。具体的には、平衡データ構造のインスタンスが独立して修正された場合、各インスタンス内でデータアイテムをグループに分割する区切り点が不一致になる場合がある。その結果、異なる平衡データ構造インスタンスのグループ又は下位グループ間のデータアイテムメンバーシップに関して、直接の一致がない場合がある。２つの当該インスタンスを調整するためには、２つのインスタンスの全体を徹底的に比較することが必要であることがあり、各インスタンスインデックスが多数のデータアイテムである場合は非常に時間がかかる場合がある。

【０１６４】

数量に従いデータアイテムをグループ間に分散する平衡データ構造の代替として、一部の実施形態において、キーマップインスタンス１４０のインデックスデータ構造は、各グループ内のデータアイテム間の一部の関係に従い、グループ間でデータアイテムを分散する、非平衡データ構造（トライとしても言及される）を実装するように設定されてもよい。具体的には、キーマップインスタンス１４０は、対応するキー１４６のプレフィックスに従い、エントリ１４４をインデックス化するように設定されてもよい。例として、対応する、大文字と小文字を区別しない英数字キー１４６を有する６００のキーマップエントリ１４４が存在する事例を考慮されたい。これらの６００のエントリの平衡インデックスは、それぞれ２００のエントリの３つの平衡グループのエントリに分割されるであろう。それに反して、一実施形態において、非平衡インデックスは、ａからｌまでの文字で始まるこれらのエントリが、第１のグループに割り当てられ、ｍからｘの文字で始まるこれらのエントリが第２のグループに割り当てられ、ｙ又はｚの文字又は数字０～９で始まるこれらのエントリが第３のグループに割り当てられる、３つの英数字グループを定義するであろう。

【０１６５】

エントリ１４４は、非平衡インデックスのグループにわたって不均一に分散されてもよい。例えば、第１のグループに３００のエントリ、第２のグループに２５０のエントリ、そして第３のグループにわずか５０のエントリがある場合がある。しかしながら、いかなる所定のエントリ１４４について、非平衡の特定のグループにおける所定のエントリ１４４のメンバーシップは、いかなる特定のグループにおける多くのエントリ１４４に依存することなく、その対応するキー１４６の機能であってもよい。したがって、非均衡印デックする２つが、同一のグループ定義を維持する場合、各グループは、一方のグループに依存することなく、独立して同期化されてもよい。例えば、２つのインスタンス間におけるａ～ｌグループは、ｍからｘグループ及びｙから９グループから独立して同期化されても

よい。それに反して、上記に記載されるように、エントリ１４４の同一の組の平衡インデックスの２つのインスタンスの同期化は、すべてのグループにわたるすべてのエントリを考慮することを必要とする場合がある。

【０１６６】

多くのデータアイテムをインデックス化するための非平衡データ構造の使用を示す１つの例を、図１７に示す。図示した実施形態において、非平衡インデックス２００（又は、単にインデックス２００）は、プレフィックス「ａｌ」で始まる多くの文字値をインデックス化するための階層的な方法に配置される、多くのノード２１０ａを含む。例えば、インデックス化された値は、キーマップインスタンス１４０の様々なエントリ１４４のキー１４６と対応してもよい。インデックス２００内の各ノード２１０は、インデックス化されるデータアイテムと直接一致してよい、又は一致しなくてよい関連するタグ値を含む。図示した実施形態において、楕円で描かれたノードは、対応するデータアイテムを有しないインデックス２００の内部ノードと対応してもよいが、長方形で描かれたノードは、インデックス化されたデータアイテムと対応してもよい。したがって、例えば、ノード２１０ａは文字列「ａｌ」と対応し、インデックス２００内の多くの他のノードと関係するが、文字列「ａｌ」と対応する実際のキー１４６は存在しない場合がある。それに反して、「ａｌｉｃｉａ」というタグを有するノード２１０ｎは、同一の文字列を特定するキー１４６と対応してもよい。内部ノード２１０と非内部ノード２１０との間の相違は、ノード２１０の状態を明示的に反映してよく、又は反映しなくてもよい。

【０１６７】

以下に説明するように、一部の実施形態において、非平衡データ構造は、他の複数のインデックスの１つのインデックスとして構成されてもよい。このような一部の実施形態において、インデックス２００の第１のインスタンス内のデータアイテムは、別のインデックス２００のルートノード２１０であってもよく、第１のインデックス２００内の対応するノード２１０は、非内部ノードと見なされる場合がある。すなわち、一部の実施形態において、所定のインデックス２００の非内部ノード２１０は、所定のインデックス２００の外部である、エントリ１４４、又は別のインデックス２００のルートノードなど、データ値と関連するいかなるノード２１０として概して定義される場合がある。同様に、所定のインデックス２００の内部ノードは、所定のインデックス２００内の他のノード２１０のみを参照してもよく、エントリ１４４、又は所定のインデックス２００から異なる他のインデックス２００とのあらゆる関連を担ってもよい。また、図１７に示すように、非内部ノード２１０は、リーフノード（例えば、さらに低い階層レベルで、他のノードを参照しないノード）である必要はないことに留意されたい。

【０１６８】

様々な実施形態において、各ノード２１０は、様々な情報を暗号化してもよい。ノード内でコード化されてもよい様々なデータフィールドを示す、ジェネリックノード２１０の一実施形態を、図１８に示す。具体的な実施形態において、ノード２１０は、タグフィールド２１２、カウントフィールド２１４、フィンガープリントフィールド２１６、及び１つ以上のポインタフィールド２１８を含む。概して、タグ２１２は、以下にさらに詳しく説明するように、インデックス２００をトラバースする又は操る過程で使用されてもよい、所定のノード２１０と対応する値を保存するように設定されてもよい。一部の実施形態において、タグ２１２は、インデックス２００内のすべてのノード間からノード２１０を独自に識別してもよい。また、一部の実施形態において、所定のノード２１０のタグ２１２は、インデックス２００内の所定のノード２１０のすべての直接の先祖であるタグ２１２に、プレフィックスとして含んでもよい。すなわち、所定のノード２１０のタグ２１２は、その所定のノードの直接の親ノード２１０のタグに対する一部の値を加えることを通じて決定されてもよい。例えば、タグ、「ａｌｉｃｉａ」を有する、図１７のノード２１０ｎを考慮されたい。ノード２１０ｎの直接の先祖ノード２１０ｌ、２１０ｋ、及び２１０ａのそれぞれは、ノード２１０ｎのタグの適切なプレフィックスを形成するタグ（それぞれ「ａｌｉｃ」、「ａｌｉ」及び「ａｌ」）を有する。

【０１６９】
図１７に示すように、あるノード２１０は、インデックス２００の階層のさらに下方で、１つ以上の子、又は子孫ノード２１０を参照する。一実施形態において、ポインタフィールド２１８は、所定のノード２１０から別のノード２１０へ、ポインタ、又は参照を反映するデータを保存するように設定されてもよい。例えば、所定のポインタフィールド２１８は、メモリアドレス空間などの、アドレス空間内で参照されたノード２１０の位置を識別するアドレスを含んでもよい。また、所定のポインタフィールド２１８は、参照されたノード２１０に関する付加的なタグ情報を含んでもよい。例えば、図１７に示すように、子孫ノード２１０への所定のノード２１０からの各アークは、所定のノード２１０のタグ２１２によって形成されるプレフィックスとは異なる子孫ノード２１０のタグ２１２の第１の文字をラベルされる。一実施形態において、この付加的なタグ情報は、参照されたノード２１０に対するポインタと共に、対応するポインタフィールド２１８内に保存されてもよい。例えば、ノード２１０ａに含まれるポインタフィールド２１８は、対応するタグデータ、「ａ」、「ｅ」、「ｆ」、「ｉ」、及び「ｚ」のみならず、ノード２１０ｂ、２１０ｇ、２１０ｊ、２１０ｋ、及び２１０ｔに対する参照をそれぞれ含んでもよい。
【０１７０】
図１２に関連して上記に記載されるように、インデックス２００のようなインデックスは、選択のためにキーマップエントリ１４４などのデータアイテムを体系化するために使用されてもよい。一部の実施形態において、非内部ノード２１０のポインタフィールド２１８（すなわち、インデックス化されたデータアイテムに直接マップするノード２１０）も、キーマップエントリ１４４、ブロック４２５、又はブリック４１５などの、対応するデータアイテムに対するポインタを含んでもよい。一部の実施形態において、以下にさらに詳しく説明するように、インデックス２００のような、非平衡インデックスは、１つのインデックス２００の非内部ノード２１０が、別のインデックス２００を参照してもよいような、階層的に実装されてもよい。インデックス化されたデータアイテムを参照するポインタフィールド２１８は、ポインタフィールド２１８の異なる種類のために異なるコード化を使用することなどの、あらゆる適切な技術によって、別のノード２１０を参照するポインタフィールド２１８から区別されてもよい。例えば、前の段落内で説明したように、子孫ノード２１０に対するアークに関連するタグ情報がポインタフィールド２１８内でコード化された実施形態において、空タグは、子孫ノード２１０への参照から、インデックス化されたデータへの参照を区別するために使用されてもよい。
【０１７１】
所定のノード２１０について、カウントフィールド２１４及びフィンガープリントフィールド２１６は、所定のノード２１０の真下のノード２１０の状態を参照するように設定されてもよい。一実施形態において、カウント２１４は、所定のノード２１０の子孫（例えば、階層的に間下にある）であるすべてのノードのカウントを保存するように設定されてもよい。例えば、図１７のノード２１０ｋは、その真下に、インデックス２００内で８つの他のノード２１０を有する。同様に、そのカウント２１４は、あらゆる適切なコード化又は形式を使用する、８つの値を表示してもよい。
【０１７２】
様々な実施形態において、所定のノード２１０のフィンガープリントフィールド２１６は、所定のノード２１０の階層的に真下にあるノード２１０のデータの一部分で行われたハッシュの値標示（例えば、適切なハッシュアルゴリズムの結果）を保存するように設定されてもよい。例えば、所定のノード２１０のフィンガープリントフィールド２１６は、所定のノード２１０の子孫であるすべてのノード２１０のタグ２１２のハッシュの合計を反映してもよい。代わりに、フィンガープリントフィールド２１６は、トラバーサル（例えば、横型トラバーサル、又は縦型トラバーサル）の特定の、一貫した順序に従い、子孫ノード２１０のタグ２１２の連結のハッシュを反映してもよい。他の実施形態において、タグ２１２のそばのノード２１０の他のフィールドは、ハッシュ化に関与してもよい。一部の実施形態において、所定のノード２１０と関連するデータは、それ自体のフィンガー

プリントフィールド２１６内で反映されてもよいが、他の実施形態において、所定のノード２１０のフィンガープリントフィールド２１６は、その子孫ノードに基づいて厳密に決定されてもよい。記述の一貫性について、本願で使用した所定のノード２１０のフィンガープリントは、所定のノード２１０の少なくとも一部の子孫ノードの機能であるハッシュ値を参照してもよいが、所定のノード２１０のハッシュは、子孫ではなく、所定のノード２１０のみと関連するデータの機能である、ハッシュ値を参照してもよい。

【０１７３】

一般的に言えば、ハッシュアルゴリズムは、２つのハッシュ値が異なる場合、２つのハッシュ値が作成された元のソースデータ値も何らかの方法で違わなければならない、小さい、一般的に固定された長さのハッシュ値の上に、あるいは任意の長さの所定のソースデータ値をマップするように設定されてもよい。ハッシュアルゴリズムは、一般的に１対１の機能ではなく、２つのハッシュ値間のアイデンティティは、元のソースデータ値間のアイデンティティを必ずしも暗示しない。しかしながら、ハッシュアルゴリズムの一部のクラスについて、同一のハッシュ値を与えられた元のソースデータ値間のアイデンティティは、とりわけ、冗長のある程度を示すソースデータ値について、統計的に定量化できる確率、又は信頼度内である可能性がある場合がある。ハッシュアルゴリズムの異なる種類は、署名、フィンガープリント、又はチェックサムアルゴリズムと言われることもある。ハッシュアルゴリズムのいかなる適切な種類は、あらゆる適切なバージョンのＭｅｓｓａｇｅ　Ｄｉｇｅｓｔ５（ＭＤ５）アルゴリズム、又は、ＳＨＡ－１、ＳＨＡ－２５６、ＳＨＡ－５１２などのようなＳｅｃｕｒｅ　Ｈａｓｈ　Ａｌｇｏｒｉｔｈｍ（ＳＨＡ）を含む、制限されない例を手段として、フィンガープリントフィールド２１６に保存されるハッシュ値を作成するために用いられてもよいことを意図する。

【０１７４】

前項に記載のように、キーマップインスタンス１４０で行われる基本的な操作は、操作に対するパラメータとして特定されるキーと対応するエントリ１４４をそれぞれ保存し及び取り出す、格納及び取得操作を含んでもよい。一部の実施形態において、キーマップインスタンス１４０内の様々なインデックスは、インデックス２００などの非平衡インデックスとして実装されてもよい。

【０１７５】

多数のデータアイテムをインデックス化する場合、キーマップインスタンス１４０で共通する場合があるように、すべてのデータアイテムのためのインデックス２００の１つのインスタンスを使用することは、実用的でないであろう。例えば、１つの大きなインデックスは、インデックスを処理するシステムのメモリに完全に収まらない場合があり、インデックスに左右される操作の能力にマイナスに影響を与える場合がある。一部の実施形態において、大きなインデックスは、層別の、非平衡データ構造、又は層別インデックスを使用して実装されてもよい。一般的に言えば、層別インデックス内で、インデックス２００の複数のインスタンスは、階層的に識別されてもよく、階層が上であるインスタンスは、他のインデックス２００をインデックス化してもよく、階層が下であるインデックスは、特定のエントリ１４４、又は他のエンティティ（例えば、ブロック４２５、又はブリック４１５）をインデックス化してもよい。

【０１７６】

層別インデックスの一実施形態を図１９に示す。図示した実施形態において、層別インデックス２２０は、５つのインデックス２００ａ～ｅを含む。インデックス２００ａは、ノード２１０ｕ～ｘを含み、各ノードはインデックス２００ｂ～ｅのうちの１つのそれぞれのルートノードを参照する、非内部ノードである。代わりに、インデックス２００ｂ～ｅは、図１７に示されたノード２１０ａ～ｔの様々なものをそれぞれ含む。層別インデックス２２０の一部の実施形態において、インデックス２００ａのような、上位インデックスは、メモリ、キャッシュ、又はシステム処理インデックスの別の上位メモリ階層に属するように設定されてもよいが、インデックス２００ｂ～ｅなどの下位インデックスは、ディスク、又は別の当該メモリ階層の下位に主として属してもよい。このような実施形態に

おいて、下位インデックスは、例えば、ページングタイプの技術を使用して、必要があれば、メモリ階層の下位から上位に再配置されてもよい。多数のデータアイテムのインデックスの階層的なパーティションをサポートすることによって、層別インデックス２２０は、システム供給源をさらに効率的、効果的に使用できる。

【０１７７】

例えば、前述のページング技術を使用して、層別インデックス２２０の頻繁に使用されたインデックス２００は、容量が限られるが一般的にアクセスすることが速い上位のメモリ階層に保存されてもよいが、それほど頻繁に使用されないインデックス２００は、上位よりは大きなストレージ容量を有するが一般的にアクセスすることが遅い下のメモリ階層位に保存されてもよい。一部の実施形態において、ノード２１０が、層別インデックス２２０内でインデックス２００に加えられると、個々のインデックス２００は、目的とするサイズ（インデックスに実装するシステム上のディスクブロック、又はメモリページなど）を超えて増大する場合があることを考慮されたい。このような実施形態において、所定のインデックス２００が目的とするサイズを超えて増加した場合、２つ以上のインデックスインスタンスに分割されてもよい。当該の分割の実行過程において、ノード２１０は、新規インデックスインスタンスを説明するために必要であるものとして上位インデックス２００に加えられてもよい。

【０１７８】

キーマップエントリの格納又は取得操作に応じて、層別、又は非層別非平衡インデックスは、特定のキーが、インデックス２００内のノード２１０と対応するかどうかを決定するためにトラバースされてもよい。非平衡インデックストラバーサルの方法の一実施形態を図２０に示す。図示した実施形態において、インデックス内でキー値を検索する（検索値としても言及される）ブロック２０００で開始する操作を、例えば、関連するキーマップ操作を介して、特定する。それに続き、インデックスのルートノード２１０（例えば、親ノードを有さないノード２１０）が選択される（ブロック２００２）。

【０１７９】

選択されたノード２１０について、ノードの対応するタグ値２１２は、タグ値が、検索値と正確に一致するか、検索値のプレフィックスであるか、又はそのいずれでもないかどうかを決定するために検索値に対して比較される（ブロック２００４）。選択されたノード２１０のタグ値２１２が検索値と一致する場合、選択されたノード２１０は、内部、又は非内部ノードであるかどうか決定するために調査される（ブロック２００６～２００８）。例えば、ポインタ２１８又は選択されたノード２１０の他の内容は、ノードが、エントリ１４４、又はインデックス２００の別のインスタンスなどのインデックス２００によってインデックス化されたデータ値を参照するかを決定するために調査されてもよい。選択されたノード２１０が内部ノードである場合、以下に説明するようにインデックス不足が生じる場合がある（ブロック２０２２）。

【０１８０】

選択されたノード２１０が非内部ノードである場合、選択されたノード２１０によって参照されたデータ値は取り出される（ブロック２０１０）。層別非平衡データ構造をサポートする実施形態において、一部のデータ構造インスタンスは、他のデータ構造インスタンスをインデックス化してもよく、取り出されたデータ値はエントリ１４４、又はインデックス２００の別のインスタンスのルートノードのいずれかと対応してもよい。取り出されたデータ値がエントリ１４４である場合、インデックストラバーサルが完了され、取り出されたエントリ１４４は、トラバーサルを始めたキーマップ操作に従い、処理されてもよい（ブロック２０１２－２０１４）。例えば、開始しているキーマップ操作が取得操作の場合、取り出されたエントリ１４４が取得操作の結果として返されてもよい。例えば、開始しているキーマップ操作が格納操作の場合、取り出されたエントリ１４４が、格納操作で特定されたパラメータに従い修正されてもよい。

【０１８１】

取り出されたデータ値が、エントリ１４４と対応しない場合、図示した実施形態におい

て、それは別のインデックス２００のルートノード２１０と対応してもよい。同様にこのルートノード２１０が選択されてもよく（ブロック２０１２、２０１６）、操作は、新規に選択されたインデックス２００のトラバーサルであるブロック２００４から処理されてもよい。したがって、一実施形態において、図２０の方法の実行は、検索値と対応するノード２１０の存在又は不足が断定的に決定されるまで続いてもよい。

【０１８２】

ブロック２００６に戻って、選択されたノード２１０のタグ２１２が検索値と一致しないが、検索値のプレフィックスである場合、選択されたノード２１０の子孫は、どの子孫も検索値と対応するかを判断するために調査されてもよい（ブロック２０１８）。そうである場合、対応する子孫ノード２１０は選択されてもよく（ブロック２０２０）、操作はブロック２００４から処理されてもよい。一実施形態において、選択されたノード２１０のポインタ２１８は、選択されたノード２１０のタグ２１２と共に取得された時、付加的なタグ情報が特定のポインタ２１８と関連しているか、また検索値のプレフィックスを形成する（又は完全に一致する）かどうか決定するために調査されてもよい。例えば、図１７に言及して、ノード２１０ａのタグ「ａｌ」は、「ａｌｉｂａｂａ」の検索値のプレフィックスであると決定されてもよい。さらに、対応するポインタ２１８によって代表されてもよい、ノード２１０ａからノード２１０ｋのアークは、付加的なタグ情報「ｉ」と関連する。このタグ情報は、ノード２１０ａのタグ「ａｌ」に添付されると、検索値のプレフィックスでもある、値「ａｌｉ」を形成する。したがって、ノード２１０ｋは、将来のトラバーサルのために選択されてもよい。

【０１８３】

ブロック２０１８に戻って、検索値と対応する選択されたノード２１０の子孫がない場合、検索値は、インデックス２００内で対応するエントリ１４４を持たず、インデックス不足としても言及される（ブロック２０２２）。該インデックス不足は、インデックストラバーサルを始めたキーマップ操作の種類に従い処理されてもよい（ブロック２０２４）。例えば、キーマップエントリ取得操作は、要求するクライアントに不足の適切な状態表示標示を返すことによって処理してもよい。対照的に、キーマップエントリ格納操作は、選択されたノード２１０の子孫としてインデックスに保存されるエントリ１４４と対応する新規ノード２１０を挿入することによって、インデックス不足を処理してもよい。例えば、新規ノード２１０が作成されてもよく、その様々なフィールドが、保存されるエントリ１４４のために適切に設定されてもよく、新規ノード２１０に対するポインタ２１８が、選択されたノード２１０内に保存されてもよい。新規ノード２１０がインデックス２００に付加された場合、又は存在するノード２１０が修正された場合、付加された、又は修正されたノード２１０のすべての先祖ノード２１０のカウントフィールド２１４及びフィンガープリントフィールド２１６は、変更を反映するために更新されてもよい。

【０１８４】

ブロック２００６に戻って、選択されたノード２１０のタグ２１２が検索値と一致せず、検索値のプレフィックスである場合、インデックス不足が生じる場合があり、ブロック２０２２から処理が続けられる。一部の例において、この事例は、選択されたノード２１０がインデックス２００のルートノードである時に生じる場合がある。同様に、一実施形態において、この新規ノード２１０をインデックス２００に付加することは、検索値及び存在するルートノード２１０（この場合、選択されたノード２１０）のタグ２１２の両方に共通のプレフィックスであるタグ２１２を有する新規ルートノード２１０を作成することを含む。（一部の例において、新規ノード２１０の共通のプレフィックスは、あらゆる値のための有効なプレフィックスとして解釈される、空値である場合がある。）新規ルートノード２１０は、子孫として、選択されたノード２１０を参照するように設定されてもよい。必要な場合、付加的なノード２１０は、検索値と対応し、新規ルートノード２１０の付加的な子孫として構成されるために作成されてもよい。

【０１８５】

一部の実施形態において、インデックスの不足は、選択されたノード２１０のタグ２１

２が検索値と一致せず、検索値のプレフィックスでない場合、階層的非平衡インデックス２００をトラバースする間、ただちに生じなくてもよいことに留意されたい。一実施形態において、これらの事例に遭遇した場合、選択されたノード２１０が親を有する場合、親ノード２１０が選択される。親ノード２１０が別のインデックス２００を参照する非内部ノードである場合、参照されたインデックス２００のルートノード２１０が選択されてもよく、処理はブロック２００４から続いてもよい。そうでなければ、インデックスの不足が生じる場合がある。（しかしながら、この事例は他のインデックス２００をインデックス化しない非階層的、内蔵型インデックス２００には生じなくてもよいことに留意されたい。）この事の例として、検索値が「ａｌｉｃｅ」である図１９の階層的インデックスを考慮されたい。インデックス２００ａのトラバーサルが、タグ「ａｌｉ」を有するノード２１０ｗに処理されてもよい。ノード２１０ｗは、「ａｌｉ」と共に検索値のプレフィックスを形成する、関連するタグ情報「ｃ」をもつ子孫ノード２１０ｘに対するポインタを有するため、ノード２１０ｘが選択されてもよい。しかしながら、ノード２１０ｘのタグは、一致しない、検索値のプレフィックスでない、「ａｌｉｃｉａ」である。したがって、トラバーサルは、インデックス２００ｃを参照する非内部ノードである、ノード２１０ｗ（ノード２１０ｘの親）に戻ってもよい。同様に、トラバーサルは、ノード２１０ｋに対して、そして最終的に検索値と一致するタグ２１２を有する、ノード２１０ｍに対して続いてもよい。

【０１８６】

様々な実施形態において、非平衡インデックス２００、又は階層的非平衡インデックス２２０は、キーマップインスタンス１４０内のキーマップエントリ１４４をインデックス化するために使用されてもよい。例えば、階層的インデックス２２０は、１つ以上のパーティションインデックス４１０、ブロックインデックス４２０、又はエントリインデックス４３０、又はキーマップインスタンス１４０内に実装される、あらゆる他のインデックスのレベルを実装するために用いられてもよい。上記に論考のように、異なるキーマップインスタンス１４０は、緩やかな同期プロトコルが用いられる場合、普通の操作の方法において相違、又は不一致になってもよい。一部の実施形態において、キーマップインスタンス１４０は、インデックス構造、又はインデックス化された内容の相違を識別するために、一貫した順序で（例えば、縦型、又は横型検索順序）、それぞれのインデックスデータ構造の各ノードをトラバースする、包括的なプロトコルを使用して同期化されてもよい。しかしながら、インデックスデータ構造内の、キーの数、及びカウントの含有及び／又は累積したハッシュ情報よりは、むしろキー情報に従ったデータの分散などの、上記に記載される非平衡インデックスの様々な特徴は、さらにコンピュータ的に効率的な同期化アルゴリズムの実装を容易にしてもよい。

【０１８７】

前に説明したアンチエントロピー設定の調整プロトコルの多くの可能なバージョンは、キーマップインスタンス１４０によって実装された、非平衡で、多分満たされたインデックスの使用を意図する。このようなプロトコルの一実施形態の説明は、例えば、他の事例にわたりある事例を最適にするために選択することで、又はプロトコルの一般的なステップを行うために、アルゴリズムの１つ、又は別の特定の種類又はクラスを使用することを選択することで、一般的なプロトコルの予期される変化は、異なる実装の優先順位を提示し得ることが既知であるが、以下のとおりである。したがって、説明された実施形態は、制限することよりはむしろ実例として意図されたものである。

【０１８８】

一実施形態において、非平衡インデックス２００、又は階層的非平衡インデックス２２０の異なるインスタンスを調整するように構成されるアンチエントロピープロトコルは、様々な種類のメッセージのインスタンス間の交換を含んでもよい。アンチエントロピープロトコルの一実施形態に基づいてもよい例示的なメッセージの１組は、ＤＡＴＡメッセージ、ＲＥＱＵＥＳＴメッセージ、ＨＡＳＨメッセージ、ＦＩＬＴＥＲメッセージ、及びＦＩＮＧＥＲＰＲＩＮＴメッセージを含んでもよい。これらの各メッセージのそれぞれの実

施形態の一般的な機能は、メッセージが、アンチエントロピープロトコルの実施形態を実装するためにどのように使用されてよいかの論考に続いて、以下に説明される。以下の論考において、当該キーマップインスタンスが、上記に記載された特徴のどれかを含む、非平衡インデックス２００の１つ以上のインスタンス、又は層別非平衡インデックス２２０を実装してもよいことが理解されるが、参照が、キーマップインスタンス１４０間のデータの交換になされてもよい。

【０１８９】

ＤＡＴＡメッセージは、１つのキーマップインスタンス１４０から別のキーマップインスタンスへ、１つ以上のインデックスノード２１０においてのデータを伝えるために使用されてもよい。一実施形態において、ＤＡＴＡメッセージは、所定のノード２１０と関連するタグ２１２のみを伝えるように設定されてもよいが、他の実施形態において、ＤＡＴＡメッセージは、所定のノード２１０と関連する他のフィールドを伝えてもよい。一部の実施形態において、所定のノード２１０が非内部ノードである場合、ＤＡＴＡメッセージは、所定のノード２１０と関連するデータアイテムのすべて、又は一部分を含んでもよい（例えば、エントリ１４４、又は別のインデックス２００のルートノード２１０についての情報）。

【０１９０】

ＨＡＳＨメッセージは、所定のノード２１０のフィールド、又は所定のノード２１０と関連するデータアイテムを明示的に伝えることなく、１つのキーマップインスタンス１４０から別のキーマップインスタンスへ、１つ以上のインデックスノード２１０についての情報を伝えるために使用されてもよい。一実施形態において、ＨＡＳＨメッセージは、適切なアルゴリズムに従い計算された所定のノード２１０のハッシュのみならず、所定のノード２１０と関連するタグ２１２を伝えるように設定されてもよい。一部の実施形態において、所定のノード２１０のハッシュは、所定のノード２１０と関連するデータアイテム（例えば、キーマップエントリ１４４）を反映してもよいが、所定のノード２１０のあらゆる子孫を除外してもよい。

【０１９１】

ＲＥＱＵＥＳＴメッセージは、１つ以上のノード２１０に関連する情報のための要求を伝えるために使用されてもよい。一実施形態において、ＲＥＱＵＥＳＴメッセージは、１つ以上のタグプレフィックス値を伝えるように設定されてもよい。それに応じて、要求するインスタンスは、伝えられたタグプレフィックス値が実際にプレフィックスである、タグ２１２を有するこれらのノード２１０についての情報を受け取ることを期待してもよい。所定のノード２１０について、受信した情報は、所定のノード２１０の対応するフィールドの内容及び／又は所定のノード２１０と対応するデータアイテム（例えば、キーマップエントリ１４４）を含んでもよい。一部の実施形態において、特定のタグプレフィックス値によって定義された結果空間内の値、又は値の範囲などは、タグプレフィックス値のために返された結果から除外されるべきであることを特定することによって、ＲＥＱＵＥＳＴメッセージは、要求されたプレフィックス値のさらなる必要条件をサポートしてもよい。例えば、ＲＥＱＵＥＳＴメッセージは、プレフィックス「ａｌｅｘｅ」又は「ａｌｅｘｊ」とこれらのノード２１０を除き、タグプレフィックス値「ａｌｅｘ」と一致するすべてのノード２１０についての情報が返されるべきであることを特定してもよい。

【０１９２】

今説明したメッセージは、個々のノード２１０の粒度のレベルで一般的に行われてもよい。しかしながら、キーマップインスタンス１４０間の相違が概して小さい場合（例えば、少数のノード２１０に伝えられた）、同期化過程が、一度に複数のノード２１０の状態を素早く確かめるために容易にしてもよい。一実施形態において、ＦＩＮＧＥＲＰＲＩＮＴ及びＦＩＬＴＥＲメッセージは、ノード２１０の総計についての情報を伝達するように設定されてもよい。とりわけ、一実施形態において、ＦＩＮＧＥＲＰＲＩＮＴメッセージは、１つのキーマップインスタンス１４０から別のキーマップインスタンスへ、タグ２１２とともに、ノード２１０のフィンガープリントフィールド２１６を伝えるように設定さ

れてもよい。上記に記載されるように、所定のノード２１０のフィンガープリントフィールド２１６は、所定のノード２１０の子孫の機能として決定されるハッシュ値を保存するように設定されてもよい。したがって、異なるキーマップインスタンス１４０のそれぞれのノード２１０のフィンガープリントフィールド２１６が等しい場合、それぞれのノード２１０の子孫の配列及び内容が同一であることがほぼ確実であり得る（使用されたハッシュアルゴリズムの特徴によって）。すなわち、それぞれのノード２１０から下に向かっていくキーマップインスタンス１４０の部分が同期化されることは、ほぼ確実である。

【０１９３】

フィンガープリントの使用は、相当数のノード２１０を含むキーマップインスタンスの１４０が同期化されたか否かの素早い判断を可能にする。しかしながら、対応する一部が同期化されてないことを示しているフィンガープリントは、概してどのように該部分が異なるかについてのさらなる詳細を提供しなくてもよい。一実施形態において、ＦＩＬＴＥＲメッセージは、第１のキーマップインスタンス１４０から、第２のキーマップインスタンス１４０へ、特定のプレフィックス値に対応するノード２１０の数をコード化するフィルタ値を伝えるように設定されてもよい。第２のインスタンスは、ある場合は、どの第２のインスタンスのノード２１０が、第１のインスタンスに存在していないかを確認するために、プレフィックス値に対応する自体のノード２１０をテストするための受信したフィルタ値を使用してもよい。

【０１９４】

一実施形態において、データ値の１組をフィルタ値に回復可能にコード化するあらゆる適切なフィルタリング技術が採用されてもよいことが考慮されるが、ＦＩＬＴＥＲメッセージから伝えられたフィルタ値は、Ｂｌｏｏｍフィルタであってもよい。一般的に言えば、値の１組のＢｌｏｏｍフィルタ（例えば、ノード２１０）は、Ｍが整数である、Ｍビットと対応してもよい。Ｂｌｏｏｍフィルタに、あらゆる値がコード化される前は、最初の値はゼロであってもよい。すなわち、フィルタのすべてのビットは、アサート停止状態であってもよい。Ｂｌｏｏｍフィルタは、それぞれが範囲［０、Ｍ－１］内の値でコード化されるべき値をマップする、フィルタ内でコード化すべき各値を、ｋ独立ハッシュ機能の各１組に通すことによって投入されてもよい。各ｋの結果として得られるハッシュ値について、Ｂｌｏｏｍフィルタ内で対応するビットはアサートされる（例えば、論理１値に設定する）。Ｍ及びｋは、偽陽性（以下に論考）の所望の可能性のみならず、Ｂｌｏｏｍフィルタ内でコード化されるべき値の数及び種類に従い、設計パラメータとして選択されてもよい。例えば、８のハッシュ機能を使用する１０２４ビットＢｌｏｏｍフィルタにおいて、各ハッシュ機能は、アサートされるべきフィルタの１０２４ビットの特定の１つを特定する対応する１０ビットハッシュ値を生成してもよい。

【０１９５】

所定の値がＢｌｏｏｍフィルタにコード化されたかどうかをテストするために、値は、フィルタをコード化するために使用されたｋ独立ハッシュ機能の同一の組を通り、フィルタ値の結果として得られたｋビットが調査される。フィルタの結果として得られたｋビットのどれもアサートされない場合、テスト値はフィルタ内で全くコード化されない。すべてのフィルタの結果として得られたｋビットがアサートされる場合、テスト値はフィルタ内でコード化されても、されなくてもよい。すなわち、テスト値は、フィルタ内で最初にコード化されているか、又は偽陽性であってもよい。一部の実施形態において、ハッシュ機能は、所定の値の１組が、フィルタ内で無事にコード化される時に同一の偽陽性値が作成される可能性を低くするために、Ｂｌｏｏｍフィルタが作成された各別々の場合において無作為に、又は自然発生的に作成された（例えば、現在のシステム時間の機能として）ソルト、又はシード値でパラメータ化されてもよい。

【０１９６】

したがって、例えば、第１のキーマップインスタンス１４０は、Ｂｌｏｏｍフィルタの中のプレフィックスＰと対応するノード｛Ａ，Ｂ，Ｃ，Ｄ，Ｅ｝の組をコード化してもよく、ＦＩＬＴＥＲメッセージを使用する第２のキーマップインスタンス１４０に対してフ

ィルタを伝えてもよい。第２のキーマップインスタンス１４０において、ノード｛Ａ，Ｂ，Ｘ，Ｙ，Ｚ｝の１組は、プレフィックスＰと対応してもよい。第２のキーマップインスタンス１４０は、フィルタに対する各ノードをテストしてもよく、ノードＡ、Ｂ、及びＸがフィルタでコード化されたが、ノードＹ及びＺはフィルタで全くコード化されなかったことを決定してもよい。したがって、第２のキーマップインスタンス１４０は、第１のキーマップインスタンス１４０にノードＹ及びＺが存在しないことを正しく結論づけてもよく、ノードＡ、Ｂ及びＸが、Ｘが偽陽性である、第１のキーマップインスタンス１４０におそらく存在することを結論付けてもよい。その結果、第２のキーマップインスタンス１４０は、ノードＹ及びＺについての情報を、第１のキーマップインスタンス１４０に伝えるための行動をとってもよい。

【０１９７】

ＤＡＴＡ、ＨＡＳＨ、ＲＥＱＵＥＳＴ、ＦＩＮＧＥＲＰＲＩＮＴ及びＦＩＬＴＥＲメッセージが実装され、あらゆる適切なプロトコル、又はＡＰＩに従い伝えられてもよく、メッセージを解読及び適切に処理するために必要なあらゆる付加的な情報のみならず、上記に記載されるように、情報を伝えるように構成されるフィールド、又はパラメータの様々な種類を含んでもよいことを考慮されたい。一実施形態において、メッセージは、メッセージに含まれる所定のタグ値について、送信するキーマップインスタンスが、それぞれ、得られたデータ、及び必要なデータパラメータとして言及される、対応するデータ、又は必要な対応するデータのいずれかを有するかどうかを示す、付加的なパラメータを含んでもよい。例えば、キーマップインスタンス１４０が、タグ「ａｌ」と、いくつかの数の子孫を有するノード２１０のためにＦＩＮＧＥＲＰＲＩＮＴメッセージを送信する場合、インスタンスは、「ａｌ」によって定義されたプレフィックス空間内にいくつかのノード２１０を有することを示す、得られたデータパラメータを含んでもよい。またインスタンスは、例えば、「ａｌ」によって定義されたプレフィックス空間のコピーが、不完全であると考えられる場合、必要なデータパラメータを含んでもよい。一部の実施形態において、ＤＡＴＡ又はＨＡＳＨメッセージは、必要なデータパラメータを明示的に特定してもよい一方、ＦＩＬＴＥＲ又はＲＥＱＵＥＳＴメッセージは明示的に得られデータパラメータを特定するが、得られたデータパラメータは、ＤＡＴＡ及びＨＡＳＨメッセージに潜在してもよいが、必要なデータパラメータは、ＦＩＬＴＥＲ及びＲＥＱＵＥＳＴメッセージに潜在してもよい。一実施形態においてＦＩＬＴＥＲメッセージは、必要なデータパラメータ、又は得られたデータパラメータの少なくとも１つを特定することが要求されてもよい。

【０１９８】

一実施形態において、２つのキーマップインスタンスによって実行されたアンチエントロピープロトコルは、２つのインスタンスが互いに接点を確率する時に開始してもよい。各インスタンスは、その両方がいくつかのデータを持ち、欠くことを仮定してもよい。同様に、各インスタンスは、インスタンスのルートノード２１０のタグ２１２及びフィンガープリント２１６を特定し、得られたデータ及び必要なデータパラメータを含む、ＦＩＮＧＥＲＰＲＩＮＴメッセージを他のインスタンスへ送信してもよい。例えば、層的非平衡インデックス２２０を用いるキーマップインスタンス１４０の実施形態において、ルートノード２１０は、親ノード、又は上位のインデックス２００を持たないインデックス２００内に、親ノードを持たないノード２１０と対応してもよい。

【０１９９】

ＦＩＮＧＥＲＰＲＩＮＴメッセージ処理の方法の一実施形態を、図２１に示した。図示した実施形態において、操作は、ＦＩＮＧＥＲＰＲＩＮＴメッセージが、メッセージ送信側から受信されたブロック２１００で開始する。例えば、第１のキーマップインスタンス１４０は、タグ値、フィンガープリント、及び１つ以上の得られたデータ、又は必要なデータパラメータを含むＦＩＮＧＥＲＰＲＩＮＴメッセージを、第２のキーマップインスタンス１４０に伝えてもよい。ＦＩＮＧＥＲＰＲＩＮＴメッセージが受信された後、メッセージ受信部のインデックスは、受信したタグ値が、対応するタグフィールド２１２のプレフィックスである（又は正確に一致する）ノード２１０が存在するかどうかを識別するた

めに、トラバースされてもよい（ブロック２１０２）。例えば、キーマップインスタンス１４０のインデックスは、図２０の方法を使用する、又は本願の適切な改良型を使用するルートノードから始まってトラバースされてもよい。

【０２００】

受信したタグ値がプレフィックスでない、又はいかなるノード２１０のタグフィールド２１２の正確な一致でない場合、ＦＩＮＧＥＲＰＲＩＮＴメッセージによって参照されたノードと対応するノード２１０は、メッセージ受信部に存在しなくてもよい。同様に、受信部は、最初に受信したＦＩＮＧＥＲＰＲＩＮＴメッセージに含まれるタグ値を特定する送信側に、ＲＥＱＵＥＳＴメッセージをメッセージ送信側に伝えることによって応答してもよい（ブロック２１０４）。一実施形態において、ＲＥＱＵＥＳＴメッセージの処理は、以下に詳しく説明するように続行されてもよい。一部の実施形態において、ＲＥＱＵＥＳＴメッセージは、受信されたＦＩＮＧＥＲＰＲＩＮＴメッセージが得られたデータパラメータを示す場合のみに伝えられてもよい。

【０２０１】

一部の実施形態において、アンチエントロピープロトコルの操作の間に交換された個々のメッセージの完了は、付加的なメッセージが、所定のメッセージが無事に完了することに応じて作成されたかどうかに左右される。すなわち、一部の実施形態において、個々のメッセージを処理することは、他のメッセージに関して、処理状況を把握しない、非同期方法で生じてもよい。本願に記載される例示的な実施形態の論考において、この処理状況を把握しない、非同期モデルが仮定される。したがって、ＲＥＱＵＥＳＴメッセージが作成された後、ＦＩＮＧＥＲＰＲＩＮＴメッセージの処理は、それ自体が完了と見なされてもよい（ブロック２１０６）。しかしながら、このモデルは、アンチエントロピープロトコルの一般的な操作にとって絶対に必要ではなく、他の実施形態において、いかなる所定のメッセージは、下位に作成されたメッセージとの、又は所定のメッセージへの応答において、同期化を遮断、待機、又は維持してもよい。例えば、明示的なハンドシェイク、肯定応答、再試行、又は他の種類のプロトコルは、１つのメッセージから別のメッセージへの完了の状態を伝えるために、一部の実施形態に用いられてもよい。

【０２０２】

受信されたタグ値が、メッセージ受信部で特定のノード２１０のタグ２１２のプレフィックス、又は一致として対応する場合は、受信されたフィンガープリント値は、２つのフィンガープリントが一致するかどうかを決定するために、特定のノード２１０のフィンガープリントフィールド２１６と比較されてもよい（ブロック２１０８）。一致する場合、メッセージ送信側とメッセージ受信側が、受信されたタグ値に関して同期化されることがほぼ確実（例えば、異なるデータから作成されているにもかかわらず、衝突する、又は同一の値を有する、２つのフィンガープリントを作るために使用する、フィンガープリントアルゴリズムの可能性に従い）である。例えば、プレフィックスとしての受信されたタグ値を有するいかなるノード２１０が、ＦＩＮＧＥＲＰＲＮＴメッセージが送信されたキーマップインスタンス１４０及び、メッセージが受信されたキーマップインスタンス１４０内と同一の状態である。したがって、ＦＩＮＧＥＲＰＲＩＮＴメッセージに応じて作成された付加的なメッセージはなくてもよく、メッセージは完了と見なされる（ブロック２１０６）。

【０２０３】

フィンガープリントが一致しない場合、メッセージ送信側とメッセージ受信側は、受信されたタグ値に関して同期化しておらず、送信側と受信側の状態が共に近くなるようにするためには付加的な作業が必要とされる場合がある。上記に記載されるように、ＦＩＬＴＥＲメッセージは、送信側があるノード２１０についての特定の情報を受信側と通信することを可能にするのに役立ってもよい。しかしながら、一部の実施形態において、妥当な偽陽性割合を保ちながらＦＩＬＴＥＲメッセージにコード化されてもよいノード２１０の数は、あるしきい値に限られてもよい。子孫ノード２１０の数が、受信されたタグ値と一致するメッセージ受信側ノード２１０でしきい値を超える場合、ＦＩＬＴＥＲメッセージ

を送信する前に付加的なＦＩＮＧＥＲＰＲＩＮＴメッセージの処理を行うことはさらに効率的であり得る。

【０２０４】

したがって、図示した実施形態において、フィンガープリントが一致しない場合、メッセージ受信側の特定のノード２１０のカウントフィールドが、ＦＩＬＴＥＲメッセージ処理のしきい値を超えるかどうか決定するために調査される（ブロック２１１０）。超える場合、メッセージ受信側は、受信されたタグ値がプレフィックスである特定のノード２１０の子に従い、受信されたタグ値と対応するインデックス範囲の一部を細分化するように設定されてもよい（ブロック２１１２）。各子ノード２１０について、メッセージ受信側は、それぞれの子ノード２１０のタグ２１２及びフィンガープリントフィールド２１６を特定して、最初のメッセージ送信側に、対応するＦＩＮＧＥＲＰＲＩＮＴメッセージを送り返すように設定されてもよい（ブロック２１１４）。さらに、例えば、特定のノード２１０の子によって示されたような、受信されたタグ値と対応するインデックス範囲の一部に格差がある場合、メッセージ受信側は、格差と対応するタグ値にたいして１つ以上のＲＥＱＵＥＳＴメッセージを送るように設定される（ブロック２１１６）。受信されたＦＩＮＧＥＲＰＲＩＮＴメッセージの処理は、そこで完了と見なされてもよい（ブロック２１１８）。一実施形態において、上記の行動に加え、受信されたタグプレフィックス値が特定のノード２１０の完全な一致である場合、特定のノード２１０に対応するＨＡＳＨメッセージは、メッセージ送信側に返されてもよい。

【０２０５】

例えば、図１７に示すように、メッセージ受信側のインデックス２００の特定のノード２１０ａは、タグ「ａｌ」、及び対応するタグ「ａｌａｎ」、「ａｌｅｘ」、「ａｌｆｒｅｄ」、「ａｌｉ」及び「ａｌｚ」を有する子を有してもよい。これは、メッセージ受信側が、「ａｌｂ」、「ａｌｃ」又は「ａｌｄ」で始まるであろうノード２１０についてではなく、「ａｌａｎ」及び「ａｌｅｘ」で始まるノード２１０についてのある情報を有することを示唆する。同時に、メッセージ受信側は、子のタグ間の格差に対するＲＥＱＵＥＳＴメッセージのみならず、ノード２１０ａの各ノードに対するＦＩＮＧＥＲＰＲＩＮＴメッセージを伝えてもよい。負のＲＥＱＵＥＳＴ構文がサポートされる実施形態において、メッセージ受信側は、特定のノードの子と対応するタグ以外のタグに対するＲＥＱＵＥＳＴメッセージを伝えてもよい。例えば、メッセージ受信側は、「ａｌａｎ」、「ａｌｅｘ」、「ａｌｆｒｅｄ」、「ａｌｉ」及び「ａｌｚ」とプレフィックスであるタグ以外のタグに対してＲＥＱＵＥＳＴメッセージを送ってもよい。

【０２０６】

特定のノード２１０のカウント値が処理しているＦＩＬＴＥＲメッセージのしきい値を超えない場合、また、受信されたＦＩＮＧＥＲＰＲＩＮＴメッセージが、得られたデータパラメータを含む場合、メッセージ送信側は、メッセージ受信側に存在しないノード２１０についての特定の情報を有してもよい。同様に、メッセージ受信側は、特定のノード２１０の子孫である各ノード２１０をフィルタ（例えば、上記載のＢｌｏｏｍフィルタ）にコード化する、ＦＩＬＴＥＲメッセージを送るように設定されてもよい（ブロック２１２０～２１２２）。例えば、図１７に言及して、特定のノードが、ノード２１０１と対応する場合、各ノード２１０ｍ～ｑをコード化するＢｌｏｏｍフィルタは、ＦＩＬＴＥＲメッセージを介して作成され、返されてもよい。図示した実施形態において、得られたデータパラメータが最初のＦＩＮＧＥＲＰＲＩＮＴメッセージに含まれていなかった場合、それぞれのＦＩＮＧＥＲＰＲＩＮＴメッセージは、ＦＩＬＴＥＲメッセージの代わりに、特定のノード２１０の各子に対して作成され、メッセージ送信側に返されてもよい（ブロック２１２４）。これらのＦＩＮＧＥＲＰＲＩＮＴメッセージは、得られたデータパラメータを含んでもよい。この事例において、以下のＦＩＬＴＥＲ又はＦＩＮＧＥＲＰＲＩＮＴメッセージの作成のいずれかは、受信されたＦＩＮＧＥＲＰＲＩＮＴメッセージの処理が完了してもよい（ブロック２１１８）。

【０２０７】

ＦＩＬＴＥＲメッセージ処理の方法の一実施形態を、図２２に示す。図示された実施形態において、操作は、例えば、上記に記載されるように、ＦＩＮＧＥＲＰＲＩＮＴメッセージに応じて、タグ値及びフィルタ値を含むＦＩＬＴＥＲメッセージがメッセージ送信側から受信されるブロック２２００で始まる。ＦＩＬＴＥＲメッセージが受信されると、メッセージ受信側のインデックスは、図２１に関して上記の記載と同様の方法で、受信されたタグ値（例えば、受信されたタグ値がプレフィックスである、又は一致する）と対応するノード特定のノード２１０を識別するためにトラバースされる（ブロック２２０２）。一部の実施形態において、ＦＩＬＴＥＲメッセージが別のメッセージに応じて作成される場合、受信されたタグ値と対応するノード２１０は概して存在する。

【０２０８】

メッセージ受信者は、それがある場合には、フィルタ値にコード化されないノード２１０を識別するために、ＦＩＬＴＥＲメッセージに提供されたフィルタ値に対する特定のノード２１０の各子孫をテストしてもよい（ブロック２２０４）。フィルタ値にコード化されないメッセージ受信側の各ノード２１０について、対応するＤＡＴＡメッセージが、メッセージ送信側に返されてもよい（ブロック２２０６）。ＦＩＮＧＥＲＰＲＩＮＴメッセージの処理は、そこで完了と見なされてもよい（ブロック２２０８）。上記に記載されるように、ＦＩＬＴＥＲメッセージに用いられるフィルタアルゴリズムの種類及び形態によって、偽陽性が生じる場合がある。すなわち、メッセージ受信側は、実際にはそうでない時に、ノード２１０のあるものがフィルタ値にコード化されており、したがって、メッセージ送信側と同一の状態を呈すると誤って結論付ける場合がある。したがって、アンチエントロピープロトコルの１ラウンドは、各ノード２１０に関して同期化になる２つのキーマップインスタンス１４０の結果にならなくてもよい。しかしながら、多くの実施形態において、アンチエントロピープロトコルの１ラウンドは、インスタンスがさらに相違するようになることを引き起こさなくてもよく、異なるインスタンス、及び使用されるアルゴリズムの特徴の程度によって、（例えば、フィルタのコード化のためのしきい値に定められた偽陽性の確率）プロトコルの反復のアプリケーションが、可能性のある程度を伴ういくらかのラウンド内で収束することが期待されてもよい。

【０２０９】

一部の実施形態において、ＨＡＳＨ、ＲＥＱＵＥＳＴ、及びＤＡＴＡメッセージの処理は、ＦＩＬＴＥＲ及びＦＩＮＧＥＲＰＲＩＮＴメッセージよりもかなり単純である。一実施形態において、ＨＡＳＨメッセージ受信側は、メッセージに含まれるタグ値と対応するノード２１０を識別する試みをしてもよく、識別されたノード２１０の対応するハッシュ値を計算してもよい。受信されたハッシュ値が、計算されたハッシュ値と一致する場合、識別されたノード２１０は、メッセージ送信側で対応するノード２１０とすでに同期化していてもよい。そうでない場合、受信されたタグ値を含むＲＥＱＵＥＳＴメッセージは、送信側がさらに最新のデータバージョンを取得するために返される。

【０２１０】

ＲＥＱＵＥＳＴメッセージの処理は、一実施形態において、例えば、上記に記載される非平衡インデックス誘導技術を使用して、メッセージに含まれた受信されたタグ値が一致する、又は対応するタグ値２１２のプレフィックスである各ノード２１０を識別するメッセージ受信側を、単に含んでもよい。各識別されたノード２１０について、上記に記載されるように構成された、対応するＤＡＴＡメッセージは、メッセージ送信側に返されてもよい。一実施形態において、受信されたＤＡＴＡメッセージの処理は、メッセージ受信側で、メッセージに示されたタグ値と対応するノード２１０が存在するかどうかを識別することを含んでもよい。存在しない場合、対応するノード２１０が、メッセージから抽出されたデータと共に作成され、投入されてもよい。存在する場合、存在するノード２１０及び／又はその対応するデータ値に関連するデータは、メッセージから抽出されたデータと置き換えられてもよい。一部の実施形態において、存在するノード２１０のデータは、受信されたデータがさらに最新のものである場合にのみ置き換えられてもよい。例えば、ＤＡＴＡメッセージは、メッセージ送信側でノード２１０と対応するエントリ１４４の内容

を含んでもよく、エントリ１４４は、受信されたエントリ１４４が存在するエントリ１４４よりもさらに最新であるかどうかを確かめるために、メッセージ受信側で対応するタイムスタンプ情報と比較してもよい、タイムスタンプ情報を含んでもよい。最新である場合、受信されたエントリ１４４は、存在するエントリ１４４を置き換えてもよい。

【０２１１】

図２１の一般的な同期プロトコルのバリエーションは可能であり、考えられる。例えば、一定の長さを有するパケットを使用して行われるキーマップインスタンス間の伝達の実施形態において、回線容量使用は、特定のノード２１０と対応する１つのＦＩＮＧＥＲＰＲＩＮＴメッセージというよりはむしろ、１パケット内の複数のノード２１０のための複数のＦＩＮＧＥＲＰＲＩＮＴメッセージを伝えることによって、改善されてもよい。当該のパケットを受け取るインスタンスは、送信側とさらにメッセージを交換する必要なく、送信者と一致しないインデックス２００の特定の１つを速やかに見分けることが出来てもよい。例えば、第１のＦＩＮＧＥＲＰＲＩＮＴメッセージが一致しない場合、受信側は、パケットの送信側に対するＲＥＱＵＥＳＴ、ＦＩＬＴＥＲ、又は他のメッセージを発行する前に、パケット内で、他のＦＩＮＧＥＲＰＲＩＮＴメッセージを考慮してもよい。そうすることで、受信者は、相違をデータの特定の部分に絞ることが出来、すでに同期化されているデータ構造の他の部分に関するメッセージを交換するための不必要なネットワークトラフィックを削減してもよい。

【０２１２】

概して、アンチエントロピープロトコル及び／又は更新伝播プロトコルを使用するキーマップインスタンス調整の実行のための上記に記載されるいずれかの方法、又は技術は、インスタンス内のキーマップインスタンス１４０、又は個々のホスト４００のレベルで操作されるように構成されるキーマップコーディネータ過程によって、実装されてもよいことを意図する。非平衡データ構造のためのアンチエントロピープロトコルを実装するための、前述の方法及び技術の多くの変更は可能であり、企図され、上記の論考は、限定的というよりは実例であることが意図される。例えば、他と頻繁に伝達する一部のエンティティを介するプロトコルの一般的なクラス、ネットワーク全体を通じて情報を分散するために無作為に選択されたエンティティは、ゴシップ系プロトコルとして言及されてもよく、ゴシップ系のプロトコルの他の技術、又は態様は、キーマップインスタンス１４０間のアンチエントロピープロトコルの使用に用いられてもよい。様々な実施形態において、上記に記載される実例同期化メッセージ（又は、他の適切なメッセージ）は、異なる特徴を有する同期プロトコルをもたらす異なる様式に組み合わされてもよい。

【０２１３】

さらに、図１７～２２に関し、上記に記載される層的インデックス化データ構造及び同期化技術がキーマップインスタンス１４０内で使用するための効果的なデータ構造の実装の内容で論考されてきたが、当該データ構造及び同期化技術は、速やかなアクセスのためにインデックス化されてもよい、大量のデータのあらゆるアプリケーションに用いられてもよいことを意図したものである。当該のアプリケーションは、図２のシステムのような、オブジェクトストレージシステムを必ずしも含む必要はないが、データインデックス化が適用できる、データ系システム、検索システム、又はあらゆる他のアプリケーションを含んでもよい。

【０２１４】

様々な実施形態において、本願に説明された事例の無作為な作成、又は選択のいかなる種類の実装は、乱数、又は自称発生のためのいかなる適切なアルゴリズム、又は技術を用いてもよいことを留意したい。多くの場合、無作為の方法を実装するコンピュータ技術は、純粋に無作為な結果を生まず、むしろ擬似乱数の結果を生む場合がある。例えば、擬似乱数アルゴリズムは、確率的に無作為な結果を作成するように構成される決定論的な過程を特定してもよい。本願で使用されるように、「無作為」又は「実質的に無作為」データの作成は、純粋に無作為データ供給源のみならず、あらゆる適切な擬似乱数コンピュータ技術を含むことを意図するものである。

【０２１５】
ストレージサービスコンポーネント検出及び管理
ストレージサービスシステムの大規模な、極めて分散された実装において、システム全体にわたって分散された図２に示す多くの様々なシステムコンポーネントがあってもよい。例えば、ビットストアノード１６０、コーディネータ１２０、及びキーマップインスタンス１４０の数百、又は数千の例がある場合がある。このような規模の分散型システムの状態を管理することは、実用的な課題をもたらす。例えば、特定のシステムコンポーネントの異なるインスタンスは、計画されたメンテナンス、コンポーネントに依存するコンピュータ供給源の障害、機能に反してコンポーネントを孤立させる伝達障害、又は他の理由のため、いかなる時に稼動しない場合がある。さらに、新しい、又は以前のコンポーネントの非稼動は、任意の、又は予想不可能な時にある事例において稼動に戻る場合がある。
【０２１６】
一実施形態において、発見、障害、及び検出デーモン（ＤＦＤＤ）１１０は、ストレージサービスシステムの様々な関連するコンポーネントの状態をそれぞれ監視するように構成され、当該の状態に関して互いに伝達しあうように構成され、当該のクライアントが、キーマップ、又はビットストア操作などの、システム操作を実行するために使用されてもよい、利用可能なシステムコンポーネントを識別してもよいことを通じるインターフェースでＤＦＤＤクライアントアプリケーションを提供するように設定されてもよい。概して、ＤＦＤＤ１１０は、他のコンポーネントに代わって、ストレージサービスシステムコンポーネントの現在の状態の一様にアクセス可能なビューを提供するように設定されてもよい。すなわち、他の、類似しないコンポーネントとの状態情報の直接の伝達のために構成された、複数の異なるインターフェースを有するストレージサービスシステムの様々なコンポーネントを構成するよりはむしろ、当該の情報を提供し、それに依存する各コンポーネントは、基本的なＤＦＤＤインターフェースを介してＤＦＤＤ１１０のインスタンスと伝達するように設定されてもよい。一部の実施形態において、ＤＦＤＤ１１０は、オペレーティングシステムによって管理される環境内で行われるように構成される、デーモン過程として実装されてもよい。しかしながら、他の実施形態において、ＤＦＤＤ１１０は、オペレーティングシステム、又は他のコンポーネントに依存、又は従属する必要なく、本願に記載される機能性を実装するように構成される、独立した、又は自立したハードウェア、又はソフトウェア媒体として、実装されてもよい。
【０２１７】
一般的に言えば、ＤＦＤＤ１１０のインスタンスによって発見、及び監視されるように構成される、ストレージサービスシステムコンポーネントの各インスタンスは、アプリケーションインスタンスといわれることもある。例えば、操作状態、又は所定のビットストアノード１６０のヘルス状態は、所定のビットストアノード１６０によって実行されるために構成されるＳＮＭコントローラ１６１のインスタンスによって示されてもよい。したがって、ＳＮＭコントローラ１６１は、ビットストアアプリケーションインスタンスと対応してもよい。同様に、キーマップインスタンス１４０の操作状態は、キーマップインスタンス内の１つ以上のホスト４００で実行されるために構成されるキーマップマネージャーのインスタンスによって示されてもよい。各キーマップマネージャーインスタンスは、キーマップアプリケーションインスタンスと対応してもよい。他の種類のアプリケーションインスタンスも可能であり、考えられる。例えば、一実施形態において、１つ以上のストレージサービスシステムコンポーネントが展開されたものを介する各コンピュータシステムは、プロセッサ、メモリ、ディスク、入力／出力（Ｉ／Ｏ）、又は他のシステム供給源の使用などの、システム特定の操作状態の詳細を検出し、報告するように構成される、ホスト監視アプリケーションインスタンスを含んでもよい。一部の実施形態において、ＤＦＤＤ１１０の各インスタンスは、それ自体アプリケーションインスタンスとして設定されてもよい。すなわち、ＤＦＤＤインスタンスは、他のアプリケーションインスタンスの状態に加えて、それ自体、操作状態を監視するように設定されてもよい。
【０２１８】

ストレージサービスシステム内で、アプリケーションインスタンスは、一般的にアプリケーション名で識別され、それぞれのアプリケーションインスタンス識別子（ＩＤ）によって固有に識別されてもよい。例えば、特定のアプリケーション名は、「キーマップ－マネージャー」、「ビットストアーマネージャー」、「ホスト－マネージャー」、又は別の適切な名前などの、アプリケーションインスタンスの包括的な種類を識別する文字列を含んでもよいが、アプリケーションインスタンスＩＤは、アプリケーション名前領域内で特定のインスタンスを固有に識別する文字列を含んでもよい。一部の実施形態において、アプリケーションインスタンスＩＤは、「キーマップ－マネージャー－４ＡＢ８Ｄ９４５」などの、アプリケーション名を明示的に含んでもよい。アプリケーションインスタンスＩＤのための他の適切な形式も用いられてもよい。一実施形態において、ＤＦＤＤ１１０の所定のインスタンスは、それぞれの状況情報を有する、多くのアプリケーションインスタンス（例えば、名前及びインスタンスＩＤを介して）を関連させるように設定されてもよい。例えば、図２３に示す実施形態において、ＤＦＤＤ１１０は、それぞれがアプリケーション名１１２及びインスタンスＩＤ１１３とインスタンス状況情報１１４を結びつける、多くのエントリ１１１を含む。一部の実施形態においてＤＦＤＤ１１０は、状態情報１１４と、所定のアプリケーション名１１２及びインスタンスＩＤ１１３との異なる種類の関連づけを反映するための１つ以上のテーブルを用いてもよいが、他の実施形態において、ＤＦＤＤ１１０は、ツリー、上記に記載されるような非平衡インデックス、又は所定のアプリケーションと対応する状態情報間の関連付けを暗示するあらゆる他の適切な種類のデータ構造を用いてもよい。

【０２１９】

一部の実施形態において、アプリケーションインスタンスＩＤは、粒度の任意のレベルのそれ自体の名前領域を含んでもよいことに留意されたい。例えば、一実施形態において、所定のキーマップアプリケーションインスタンスＩＤは、＜マップ名＞／＜インスタンス＞／＜エンドポイント＞形式の場合がある。＜マップ名＞という用語は、所定のキーマップ展開と概して対応してもよい、キーエントリ関連付けの特定のキーマップ辞書を識別してもよい。（キーマップアプリケーションインスタンスが、異なるキーマップ展開をＤＦＤＤ１１０の同一のインスタンス内で管理することは可能である。）＜インスタンス＞という用語は、固有の文字列によって、キーマップインスタンス１４０内の特定のホスト４００を識別してもよい。＜エンドポイント＞という用語は、識別されたホスト４００（例えば、特異的な過程）で実行する多くの機能的に同一のキーマップアプリケーションの１つを識別してもよい。アプリケーションインスタンスＩＤ内の他の複雑な名前領域は、可能であり、考えられる。

【０２２０】

ＤＦＤＤ１１０によってアプリケーションインスタンスと関連する状態情報は、様々な異なる種類の情報を含んでもよい。一実施形態において、ＤＦＤＤ１１０は、ＤＦＤＤ１１０によって管理されるすべての種類のアプリケーションインスタンスに共通であってもよい、広域状態情報である、状態情報１１４内に保存されるように設定されてもよい。例えば、さらに以下に詳しく説明するように、一部の実施形態において、ＤＦＤＤ１１０は、状態の１組の間の考えられる移行のみならず、アプリケーションインスタンスの広域操作状況（又は、単に広域状態）の１組を定義する、広域操作状態マシンを実装してもよい。このような実施形態において、ＤＦＤＤ１１０によって管理された各アプリケーションインスタンスは、いかなる時に、広域状態の組の特定の１つと関連してもよく、所定のアプリケーションインスタンスのための広域状態は、アプリケーションインスタンスのマシン、及び動作に従い、時間と共に変化してもよい。

【０２２１】

アプリケーションインスタンスの広く異なる種類に共通してもよい、広域状態情報に加えて、一部の実施形態において、状態情報１１４は、特定のアプリケーションインスタンス又はインスタンスの種類に特定されてもよく、又はカスタマイズされる場合がある操作状態情報を反映してもよい。例えば、アプリケーションインスタンスが、特定のビットス

トアノード１６０のビットストアマネージャーと対応する場合、その状態情報１１４は、その特定のノードで利用可能なストレージリソースの量、これらの供給源の該種類（例えば、高性能、低性能など）、又はビットストアノードの該内容に特有である、あらゆる他の関連する状態情報を含んでもよい。同様に、特定のキーマップインスタンス１４０のキーマップマネージャーに対応するアプリケーションインスタンスについては、その状態情報は、特定のキーマップインスタンス、使用される、又は利用可能なキーマップストレージリソース、又は、他の関連するキーマップ状態情報によって管理されるエントリ１４４の数についての情報を含んでもよい。一部の実施形態において、どのアプリケーションインスタンス特有の状態情報を対応するＤＦＤＤエントリ１１１内に含むかという選択は、ＤＦＤＤクライアントの要求に従い決定されてもよい。例えば、いくつかの選択肢から特定のビットストア又はキーマップアプリケーションを選択することにおいて、コーディネータ１２０又はノードピッカー１３０を援助するために使用できてもよい状態情報は、これらのアプリケーションインスタンスのＤＦＤＤエントリ１１１内に含まれてもよい。

【０２２２】

一部の実施形態において、アプリケーションインスタンスの状態情報１１４は、どのようにＤＦＤＤクライアントがインスタンスへアクセスするかについての情報も含んでもよい。例えば、状態情報１１４は、ＤＦＤＤクライアントが、アプリケーションインスタンスとの連絡を通じて確立してもよい、インターネットプロトコル（ＩＰ）アドレス及びポート番号を含んでもよい。一部のアプリケーションインスタンスは、ウェブサービスインターフェース、出版／購読型インターフェース、又は他の適切なインターフェースなどの他の種類のインターフェースをサポートしてもよい。このような実施形態において、出版／購読チャネルを購読するための、又はアプリケーションインスタンスとの通信を確立するために必要な別の種類の処置を行うために、状態情報１１４はＵＲＬ、又はＤＦＤＤクライアントがウェブサービスコールを実行するために必要な他の情報を含んでもよい。一部の実施形態において、アプリケーションインスタンスアクセス情報に加えて、又はその代わりに、状態情報１１４は、ストレージサービスシステム内に物理的に位置付けられている場所についての情報を含んでもよい。例えば、状態情報１１４は、特定のアプリケーションインスタンスが対応する、データセンタ３００、又は領域３１０の識別子を含んでもよい。

【０２２３】

上記に記載されるように、一部の実施形態において、ＤＦＤＤ１１０は、所定のアプリケーションインスタンスが通常に操作しているか、したがって、使用できるか、又は異常な状態にあるかどうかを一般用語で示してもよい、個々のアプリケーションインスタンスのための広域状態情報を維持してもよい。一実施形態において、ＤＦＤＤ１１０のインスタンスによって監視するように構成された各アプリケーションインスタンスは、（必ずしもではないが）多くの場合、数秒、又は数分などの通常の間隔で、ＤＦＤＤ１１０へその状態を報告するように設定されてもよい。このような報告は「ハートビート」と言われることもある。ハートビート報告は、あらゆる適切なプロトコルに従い（例えば、ＴＣＰ／ＩＰメッセージとして、ウェブサービスコールとして、又は他の基準、又は所有者メッセージプロトコルに従って）通信されてもよく、情報の内容によって様々であってもよい。最小限の例として、所定のアプリケーションインスタンスは、所定のインスタンスに対応するアプリケーション名及びアプリケーションインスタンスＩＤを単に含むＤＦＤＤ１１０に対するハートビートを提出してもよい。他の事例において、所定のアプリケーションインスタンスは、ローカル供給源利用の特定の状態などの、ハートビートにおける付加的な状態情報を含んでもよい。一部の実施形態において、アプリケーションインスタンスは、ハートビートを送信する前に、それ自体の機能状態を確認するために、あるレベルの自己診断、又は自己照合を実行するように設定されてもよいが、他の実施形態において、アプリケーションインスタンスは、いかなる自己評価に依存することなく、ハートビートを送信してもよい。

【０２２４】

一般的に言えば、予想通りに、アプリケーションインスタンスが、ＤＦＤＤ１１０へハートビートを送信している場合、それは通常に操作しているという理にかなった可能性がある。ハートビートがある期間にわたって中断されなければならない場合、アプリケーションインスタンスに異常があるという理にかなった可能性がある。図２４は、ハートビートアクティビティ及び／又は他のパラメータの機能として、各アプリケーションインスタンスのためにＤＦＤＤ１１０によって維持されてもよい広域状態マシンの一実施形態を示す。図示した実施形態において、新規のアプリケーションインスタンスは、ＮＥＷ状態でオンライン化され、例えば、間もなく操作を開始し、ＤＦＤＤ１１０のインスタンスの存在を知らせ、アプリケーション名、アプリケーションインスタンスＩＤ、及び対応するエントリ１１１を使用するために、ＤＦＤＤ１１０に必要なあらゆる他の情報を提供する。新規のアプリケーションインスタンスが安定し、通常の操作を開始すると、ＯＫ状態に入る。様々な実施形態において、ＮＥＷからＯＫ状態への移行は、時間の機能（例えば、アプリケーションインスタンスの種類に基づいた初期設定時間）、アプリケーションインスタンス自己報告、管理者介入、又はこれら、又は他の要素の組み合せであってもよい。

【０２２５】

図示した実施形態において、インスタンスのＤＦＤＤ１１０への最後のハートビートが障害しきい値T_{fail}未満であるため、アプリケーションインスタンスは、時間が経過した間、ＯＫ状態を維持してもよい。例えば、ＤＦＤＤ１１０は、対応するインスタンスから受信した各ハートビートに増加された各アプリケーションインスタンスのためのカウンタを維持してもよく、T_{fail}が経過する前にその値が変更するかどうかを確かめるためにそれぞれのカウンタ（例えば、カウントダウン時間）を監視してもよい。一部の実施形態において、以下に説明するように、当該の状態のなかで相違がある場合があるが、ＯＫ（及び、あるいはＮＥＷ）以外の広域状態は、異常な操作状態、又は障害状態として概して言及される場合がある。

【０２２６】

アプリケーションインスタンスのための最後のハートビートから時間T_{fail}が経過した場合、その広域状態は、ＩＮＣＯＭＭＵＮＩＣＡＤＯへ移行してもよい。図示した実施形態において、ＩＮＣＯＭＭＵＮＩＣＡＤＯは、アプリケーションインスタンスに異常があるが、永久に障害があることを断定的に決定されていないことを示す過渡状態として機能してもよい。例えば、アプリケーションインスタンスは、一時的に行き詰まる、又はハングアップする場合があり、ＤＦＤＤ１１０へのハートビートメッセージは、遅延される、又は失われる場合があり、又は以下にさらに詳しく説明するように、ＤＦＤＤ１１０の一例は、アプリケーションインスタンスの現在の状態に関してＤＦＤＤ１１０の別のインスタンスと同期化しない場合がある。ハートビートがＩＮＣＯＭＭＵＮＩＣＡＤＯ状態のアプリケーションインスタンスから受信される場合、インスタンスはＯＫ状態へ逆移行してもよい。一部の実施形態において、ＤＦＤＤクライアントは、自身のリスクでＩＮＣＯＭＭＵＮＩＣＡＤＯ状態にあるアプリケーションインスタンスを使用することを選んでもよい。

【０２２７】

アプリケーションインスタンスが、ＩＮＣＯＭＭＵＮＩＣＡＤＯ状態から自然に回復しない場合、インスタンスに影響しているさらに深刻な問題がある場合がある。図示した実施形態において、２つの予想される障害のシナリオが生じる場合がある。ＦＡＩＬ状態によって示すように、個々のアプリケーションインスタンスは、例えば、個々のインスタンスをホストしている根本を成すコンピュータ供給源の障害によって、分離に失敗する場合がある。あるいは、アプリケーションインスタンスは、ＮＥＴＷＯＲＫ　ＳＰＬＩＴ状態によって示すように、インスタンスとＤＦＤＤ１１０との間のネットワーク通信の喪失のために機能しない場合がある。例えば、アプリケーションインスタンスは、互いにストレージサービスシステムの部分を隔絶する通信障害のため、他ではなく、ＤＦＤＤ１１０の一部のインスタンスに対して操作可能であり、アクセス可能であってもよい。

【０２２８】

所定のアプリケーションインスタンス障害が隔絶されている、又はネットワークスプリットのためあるかどうかを確実に決定するのは困難である場合がある。一部の実施形態において、ＤＦＤＤ１１０は、アプリケーションインスタンスが、ＩＮＣＯＭＭＵＮＩＣＡＤＯ状態からＦＡＩＬ状態、又はＮＥＴＷＯＲＫ　ＳＰＬＩＴ状態へ移行すべきかどうかの決定をするための利用可能な情報の様々な種類を考慮にいれる、それぞれ発見的な基準H_{fail}及び$H_{netsplit}$を用いてもよい。例えば、該基準は、所定のアプリケーションインスタンスが、別の障害状態に移行する前に、少なくともしきい値量の時間$T_{heuristic}$にわたってＩＮＣＯＭＭＵＮＩＣＡＤＯ状態であることを必要としてもよい。さらに、該基準は、所定のアプリケーションインスタンスとしての、同一の領域３１０、又はデータセンタ３００と、供給源を共有する、又はそれに属する他のアプリケーションインスタンスもＩＮＣＯＭＭＵＮＩＣＡＤＯ、ＦＡＩＬ又はＮＥＴＷＯＲＫ　ＳＰＬＩＴ状態であるかどうかを考慮に入れてもよい。例えば、所定のアプリケーションインスタンスとしての同一のＩＰアドレス上に、又は同一の領域３１０、又はデータセンタ３００内の別のアドレス上に位置する別のアプリケーションインスタンスがＯＫである場合、所定のアプリケーションインスタンスの障害は隔絶されている可能性がある場合がある。それに反して、複数のアプリケーションインスタンスがＯＫではない場合、とりわけ、アプリケーションインスタンス状態が地理、又はネットワークトポロジーに従いクラスタ化された場合に、ネットワークスプリットのシナリオである可能性がある場合がある。一部の実施形態において、ＤＦＤＤ１１０は、障害の本質を決定するための消極的に受信した状態情報を使用することに加えて、障害の疑いのあるアプリケーションインスタンスを問い合わせるように設定されてもよい。一部の実施形態において、発見的な基準は、アプリケーションインスタンスが、あるしきい値の可能性（例えば、５０％可能性より高い、９０％可能性より高い、など）に従い、確率的に機能不全の可能性があるかどうかを決定するように設定されてもよい。

【０２２９】

発見的な基準によって、機能不全のアプリケーションインスタンスは、ＦＡＩＬ状態、又はＮＥＴＷＯＲＫ　ＳＰＬＩＴ状態のいずれかに移行してもよい。一部の実施形態において、ハートビートが受信された場合、これらの状態のいずれかからＯＫ状態に戻る移行をしてもよいが、他の実施形態において、これらの状態のいずれか、又はその両方は回復可能であってもよい。ＩＮＣＯＭＭＵＮＩＣＡＤＯ状態にあるアプリケーションインスタンスは、障害の可能性に機能的、又は回復可能であることが想定される場合があるが、ＦＡＩＬ又はＮＥＴＷＯＲＫ　ＳＰＬＩＴ状態にあるアプリケーションインスタンスは、機能不全であることが予想される場合がある（一部の実施形態において、回復の可能性を有する）。概して、ＤＦＤＤクライアントは、これらの障害状態のいずれにおいても、これらのアプリケーションインスタンスを選択することを避けてもよい。一部の実施形態において、ＤＦＤＤ１１０は、これらの障害状態のいずれにおいてのアプリケーションインスタンスについての情報を、クライアントから隠すように設定されてもよい。

【０２３０】

図示した実施形態において、アプリケーションインスタンスは、ＦＯＲＧＯＴＴＥＮ状態に進める前に、それぞれT_{clean}及び$T_{recover}$期間に、ＦＡＩＬ又はＮＥＴＷＯＲＫ　ＳＰＬＩＴ状態にとどまってもよい。例えば、ＦＡＩＬのある事例において、機能不全のアプリケーションインスタンスと関連する供給源は、回復、又は分析の目的のために一定期間保存されてもよい。可能な場合、当該の供給源（例えば、ビットストアノード１６０のストレージリソース）は、新規のアプリケーションインスタンスとして再展開するために初期化されてもよい。ＮＥＴＷＯＲＫ　ＳＰＬＩＴ状態のある事例において、機能不全のアプリケーションインスタンスのないシステム操作を続けるかどうかに関連する決定をする必要がある場合があり、機能不全がある場合、どのような回復処置がとられるべきかを決定する必要がある場合がある（例えば、残存するアプリケーションインスタンスの間でオブジェクトの複製を再生成することなど）。一部の実施形態において、機能不全のアプリケーションインスタンスは、当該の回復処置が完了するまで、ＦＯＲＧＯＴＴＥＮ状

態を通過しなくてもよい。

【０２３１】

アプリケーションインスタンスのＦＯＲＧＯＴＴＥＮ状態は、ＤＦＤＤ１１０内で明確に表されなくてもよい。むしろ、一部の実施形態において、ＤＦＤＤ１１０からのＤＦＤＤエントリ１１１などの、アプリケーションインスタンスの存在する状態情報の削除によってマークされてもよい。一般に、ある例において、アプリケーションの新規のインスタンスは、ＮＥＷ状態を介してフォーガットンインスタンスへ割り当てられた同一の供給源を使用して初期化されてもよいが、アプリケーションインスタンスは、ＦＯＲＧＯＴＴＥＮ状態から回復しなくてもよい。一部の実施形態において、ＦＯＲＧＯＴＴＥＮ状態にある間に、アプリケーションインスタンスが自然にハートビートの送信を再開しなければならない場合、ＤＦＤＤ１１０は、インスタンスが忘れられた（例えば、有効なエントリ１１１ともはや対応しない）ことを認識してもよく、インスタンスに操作を中止する、又はリセットする、又はそれ自体を再初期化することを指示してもよい。

【０２３２】

一部の実施形態において、広域状態の移行を考慮に入れる経験則及び移行時間のパラメータは、アプリケーションインスタンスの異なる種類に対して異なってもよく、これらのパラメータの一部、あるいはすべては、ＤＦＤＤクライアントによって調整されてもよいことに留意されたい。また、ＤＦＤＤクライアントは、概してＤＦＤＤ１１０のインスタンスを所定のアプリケーションインスタンスの現在の広域状態を確かめるために問い合わせるが、一部の実施形態において、ＤＦＤＤ１１０は、出版／購読状態の変更通知モデルをサポートしてもよい。例えば、ＤＦＤＤクライアントは、クライアントが、特定のアプリケーションインスタンス、又はインスタンスの組の広域状態変更のすべての、又はある種の通知を受けたいということを、購読過程を介してＤＦＤＤ１１０へ通知してもよい。このような状態変更を検出する際に、ＤＦＤＤ１１０は、購読するＤＦＤＤクライアントに対して変更のメッセージ表示を伝えてもよい。

【０２３３】

しばしば、アプリケーションインスタンスは、アプリケーションインスタンスに最も近いＤＦＤＤ１１０のインスタンスへ、ハートビート情報を送信するように設定されてもよい。例えば、一部の実施形態において、アプリケーションインスタンスが、単にホストのローカルＩＰアドレスを参照すること及びアプリケーションインスタンスＤＦＤＤ通信のために用意される既知のＩＰポートを使用することによって、ＤＦＤＤ１１０のローカルインスタンスへ直ちにアクセスできるように、ＤＦＤＤ１１０のインスタンスは、１つ以上の他のアプリケーションインスタンスをホストするように構成される各コンピュータシステムで提供されることがある。しかしながら、アプリケーションインスタンスが、他ではなく、ＤＦＤＤ１１０の一部のインスタンスへその状態を報告し、その状態を同期化するためのある取り組みがない場合、ＤＦＤＤ１１０の展開されたインスタンスは不一致になってもよい。

【０２３４】

一部の実施形態において、ＤＦＤＤ１１０のインスタンス間の不一致は、ゴシップ系のプロトコルなどのキーマップインスタンス１４０に関する上記に記載されるものと類似の同期プロトコルを使用して対処されてもよい。しかしながら、多くの事例において、ＤＦＤＤ１１０のインスタンスによって集合的に管理されるＤＦＤＤエントリ１１１の数は、キーマップインスタンス１４０によって管理されるキーマップエントリ１４４の数よりも実質的に少なくてもよい。この事例の場合、簡素化された調整プロトコルがＤＦＤＤ１１０のインスタンスを同期化するために使用されてもよい。このようなゴシップ系のプロトコルの一実施形態の操作の方法は、図２５に示す。図示した実施形態において、開始しているキーマップインスタンスとしても言及される、ＤＦＤＤ１１０の１つのインスタンスであるブロック２５００で始まる操作は、同期化のためにＤＦＤＤ１１０の別のピアインスタンスをランダムに選択する。一部の実施形態において、開始するＤＦＤＤインスタンスは、開始しているＤＦＤＤインスタンスの状態情報に従い、現在機能不全状態にある（

例えば、ＮＥＴＷＯＲＫ　ＳＰＬＩＴ）これらのＤＦＤＤインスタンスの中からピアＤＦＤＤインスタンスを時には故意に選択してもよい。開始しているＤＦＤＤインスタンスが明らかに機能不全であるピアインスタンスとの連絡及び同期化に成功する場合、明らかな機能不全からの回復が容易になってもよい。

【０２３５】

開始しているインスタンスは、そのエントリ１１１に反映されるアプリケーションインスタンスの識別する情報のハッシュ値を計算してもよい（例えば、アプリケーションインスタンス名及びＩＤ、及びおそらくエンドポイント、又は他の識別する情報をハッシュすることによって）（ブロック２５０２）。ハッシュ値は、例えば、ＭＤ５アルゴリズムなどのあらゆる適切なハッシュアルゴリズムに従い決定されてもよい。開始しているインスタンスは、現在のアプリケーションインスタンス状態情報の保存されたリストと共に、ピアインスタンスへ、計算されたハッシュ値を伝えてもよい（例えば、ハートビートカウント、広域状態情報及び／又は状態情報１１４に含まれるあらゆる他の情報）（ブロック２５０４）。状態情報のリストは、開始しているインスタンス及びピアインスタンスの両方に一貫したリストを作成する、あらゆる基準に従い保存されてもよい。例えば、該リストは、アプリケーションインスタンス名及び／又はＩＤに従い保存されてもよい。

【０２３６】

上記に記載されるように、一部の実施形態において、アプリケーションインスタンスと関連する状態情報は、ハートビートメッセージ内に含まれるハートビートカウントから派生してもよいことに留意されたい。同様に、一部の実施形態において、ＤＦＤＤインスタンスは、アプリケーションインスタンスのためのハートビートカウント情報を交換してもよく、他のＤＦＤＤインスタンスから直接に状態情報を受け取るよりはむしろ、受信したハートビートカウント情報からアプリケーションインスタンスの状態情報を派生してもよい。したがって、一実施形態において、所定のＤＦＤＤインスタンスは、その情報が特定のアプリケーションインスタンスから直接に受信されたか、又は同期プロトコルを介して別のＤＦＤＤインスタンスから非直接的に受信されたかに関わらず、受信されたハートビートカウント情報に基づいて、特定のアプリケーションインスタンスの状態を（例えば、図２４の状態マシンに従い）更新するように設定されてもよい。このような実施形態において、ＤＦＤＤインスタンス間の状態情報を操作するアプリケーションインスタンスの同期化は、同期プロトコルの操作を簡略化してもよいアプリケーションインスタンスの特定の広域操作状態（例えば、ＯＫ、ＩＮＣＯＭＭＵＮＩＣＡＤＯなど）を、直接交換することなく、ハートビート情報の同期化を伴ってもよい。

【０２３７】

状態情報のハッシュ値及びリストの受信に応じて、ピアインスタンスは、開始しているインスタンスによって行われるものと一致する方法で、それ自体のアプリケーションインスタンスの識別する情報のハッシュ値を計算し（ブロック２５０６）、結果として得られたハッシュ値と、開始しているインスタンスから受信されたハッシュ値とを比較する（ブロック２５０８）。２つの値が一致する場合、開始しているインスタンスとピアインスタンスの両方が、アプリケーションインスタンスの同一の組と一致するエントリ１１１を有することが高確率である。ピアインスタンスは、状態情報の受信したリストを走査し、適切であれば受信したリストからそのエントリ１１１を更新する（ブロック２５１０）。例えば、受信したリストのハートビートカウント、又はタイムスタンプが、ピアエントリ１１１の１つに保存されたものよりも大きい、又はより最近である場合、該ピアは、受信されたリストの状態情報からエントリ１１１を更新してもよい。一部の実施形態において、ピアインスタンスは、開始しているインスタンスからのリスト受信と同時か、それに続いてのいずれかで、同様の処理のための開始しているインスタンスへ、状態情報の独自のリストを送り返してもよい。

【０２３８】

ハッシュ値が一致しない場合、少なくとも１つのエントリ１１１で、ピア及び開始しているインスタンスに既知のアプリケーションインスタンスの組が異なることがあり得る。

同様に、ピアインスタンスは、開始しているインスタンスに既知のエントリ１１１の完全なダンプを要求してもよい（これらエントリ１１１の状態情報１１４にちょうど対立する形で）（ブロック２５１２）。ピアインスタンスは、欠けていたあらゆるエントリ１１１を加え、残存するエントリ１１１の状態を同期化してもよい（ブロック２５１４）。上記のように、一部の実施形態において、ピアインスタンスは、開始しているインスタンスからのダンプ受信と同時か、それに続いてのいずれかで、開始しているインスタンスへそのエントリ１１１の完全なダンプを送り返してもよい。

【０２３９】

一部の実施形態において、システム内に存在するＤＦＤＤ１１０の各インスタンスは、一定間隔で、説明した同期プロトコル、又は本願の適切な改良型を繰り返し実行するように設定されてもよいことに考慮されたい。例えば、プロトコルは、およそ定期的に１秒間隔、又はあらゆる他の適切な間隔で、ＤＦＤＤ１１０のインスタンスによって実行されてもよい。さらに、一部の実施形態において、ＤＦＤＤ１１０のインスタンスは、およそ同一の間隔で同期プロトコルを実行してもよいが、その時々で、ＤＦＤＤ１１０のインスタンスの部分のみがプロトコルを開始する場合がある、異なる相は、互いに関連してオフセットであってもよい。

【０２４０】

一部の実施形態において、ＤＦＤＤ１１０のインスタンスは、単にストレージサービスシステム内に定義されているアプリケーションインスタンスではなく、あらゆる分散型システム内のあらゆる種類のアプリケーションインスタンスの状態情報を調整、及び通信する操作に使用される場合があることに留意されたい。また、一部の実施形態において、異なるグループのＤＦＤＤインスタンスは、異なるアプリケーションインスタンスの状態情報を管理してもよい。このような一部の実施形態において、グループは、同一グループのメンバーであり、ＤＦＤＤ同期化の条件として、識別子が一致することを要求するＤＦＤＤ１１０のインスタンスに共通識別子を割り当てることによって、互いを識別してもよい。例えば、ストレージサービスシステムのアプリケーションインスタンスを管理するＤＦＤＤインスタンスは、ストレージサービスシステムと関連していない、その他のアプリケーションインスタンスの状態を管理するように設定されているＤＦＤＤインスタンスと異なる識別子を有してもよく、同一識別子を有するそれらのＤＦＤＤインスタンスのみが、図２５の同期プロトコルに従い、互いに情報を交換してもよい。

【０２４１】

一部の実施形態において、ＤＦＤＤグループ識別子は、同一システム内に存在するアプリケーションインスタンスの異なる設定を識別するために使用されてもよい。例えば、「製品」識別子に対応するＤＦＤＤ１１０のインスタンスのある組は、ストレージサービスシステム、又は別の分散型システムの製品バージョンを管理するために配置されてもよく、製品システムに対応するアプリケーションインスタンスの組を示す場合があり、一方、別の「テスト」識別子に対応するＤＦＤＤ１１０のインスタンスの組は、アプリケーションインスタンス及び状態の異なる組に対応するシステムのテストバージョンを管理するために配置されてもよい。一部の例では、いずれかのシステムバージョンに対応するアプリケーションインスタンス及び／又はＤＦＤＤインスタンスは、同一の基礎システムリソース上（例えば、同一コンピュータシステム上）で実行されてもよいが、それらの異なるＤＦＤＤグループ識別子の長所によって、互いに透過的に行われる場合があることに留意されたい。例えば、図２５に示されるプロトコルのような同期プロトコルの実行中に、ＤＦＤＤインスタンスは、同一グループのメンバーであるかどうかを最初に判断し（例えば、グループ識別子を交換することによって）、この判断を条件とする後続の同期化ステップを実行し、従ってグループ間のアプリケーションインスタンスの状態情報の分離を促進してもよい。

【０２４２】

前述のＤＦＤＤ１１０のインスタンスを同期化するためのゴシップ系プロトコルは、ストレージサービスシステム全域に渡る既存のアプリケーションインスタンスの動作状態の

配布を助長するだけでなく、その他のシステムコンポーネントによる新しいアプリケーションインスタンスの発見も促進する場合があることに留意されたい。例えば、一度新しいアプリケーションインスタンスが初期化され、ＤＦＤＤ１１０のインスタンスと連絡を取る（例えば、システム上で局所的に動作しているインスタンス上で新しいアプリケーションインスタンスが初期化された）と、新しいインスタンスに対応する新しいエントリ１１１が作成されてもよい。新しいエントリ１１１が作成されたＤＦＤＤ１１０のインスタンスが、ＤＦＤＤ１１０の様々なその他のインスタンスでその状態を同期化する際、新しいエントリ１１１は、システム全体に伝播されてもよい。様々な目的（例えば、新しいオブジェクト３０を保存する、又はキーマップエントリ１４０を更新する）のためのアプリケーションインスタンスを識別するために、ＤＦＤＤ１１０に問い合わせるＤＦＤＤクライアントは、あらゆる既存のものばかりでなく、新しいアプリケーションインスタンスに関する状態情報を提供されてもよい。

【０２４３】

上記に記載される実施例において、障害検出及び発見に関係するアプリケーションインスタンスの状態変更は、アプリケーションインスタンスの一部又はこれらのインスタンスを参照するＤＦＤＤクライアントの干渉なしに、システム全体に伝播されてもよい。つまり、所定のアプリケーションインスタンスは、どのようにハートビート情報をＤＦＤＤ１１０の１つのインスタンスに伝達するかのみを知っていればよい。システム内のＤＦＤＤ１１０のすべてのインスタンス、その他のアプリケーションインスタンス、又は当該の所定のアプリケーションインスタンスを起動する様々なクライアントの知識を有する必要はない。同様に、ＤＦＤＤクライアントは、システム内のその他のクライアント、又はすべてのアプリケーションあるいはＤＦＤＤインスタンスの独立した知識を有する必要はなく、クライアントは、システム内で使用可能なリソースの状態に関する合理的に最新の情報を獲得するために情報をやり取りする、ＤＦＤＤ１１０のインスタンスに依存してもよい。その他のアプリケーションインスタンスを要求せずに、アプリケーションインスタンスの状態を変更すること、又はクライアントにそのような変更を即座に知らせることを可能にすることより、ＤＦＤＤ１１０は、ストレージサービスシステムの拡張性を促進してもよい。

【０２４４】

ストレージクラス

ストレージサービスシステムの一部の実施形態において、オブジェクト３０は、それらの複製レベル、領域３１０に渡る複製の分散、複製が保存されているストレージリソースの種類、及び／又はその他のシステム機能あるいはポリシーに関して一様に取り扱われる場合がある。例えば、システムは、同一回数だけ、同一番号の異なる領域３１０に、各オブジェクト３０を複製することを試みる場合がある。しかし、異なるクライアント５０は、異なるオブジェクト３０に対して、異なるストレージ要件を有してもよい。例えば、１つのクライアント５０は、初期ストレージポリシーが提供し得る信頼性の程度よりも高い信頼性で特定のオブジェクト３０を保存する（例えば、数及び複製の分散に関して）ことを望む一方、別のクライアント５０は初期設定レベルの信頼性でさえ必要としない場合がある。あるいは、クライアント５０は、可能な信頼性の費用で、オブジェクトの複製が分散する領域３１０の数を制限することによって、オブジェクトの書き込み性能を向上することを希望する場合がある。

【０２４５】

同様に、一実施形態において、図２のようなストレージサービスシステムは、オブジェクト３０のストレージクラスをサポートするように設定されてもよい。一般的にいうと、所定のオブジェクト３０のストレージクラスは、当該所定のオブジェクト３０に関するサービスレベル合意書（ＳＬＡ）に影響を与えるあらゆるストレージサービスシステム機能又は特徴の組を指定してもよい。サービスレベル合意書は、一般的に、クライアントから受け取る、ある対価（例えば、料金又はその他の対価の適切な種類）と引き換えに、サービスプロバイダがクライアントに提供するサービスに対する保証又は期待を反映してもよ

い。例えば、ストレージサービスシステムにより管理されるオブジェクト３０のＳＬＡは、様々なレベルのオブジェクトの信頼性、可用性、アクセス機能（例えば、待ち時間、バンド幅）、料金又はサービス率、あるいはクライアントのオブジェクト３０とのやり取りにおける、あらゆるその他の測定可能な態様を指定してもよい。一部の実施形態において、ストレージクラスは、ＳＬＡ特性（例えば、以下に記載されるオブジェクトの複製の数及び配布）の特定のサブセットのみを指定してもよいが、他の実施形態において、ストレージクラスは所定のオブジェクト３０に関して、ＳＬＡ合意のすべての定義された態様を網羅する総合的なＳＬＡと直接対応してもよい。

【０２４６】

一実施形態において、ストレージサービスシステムは、それぞれが特定の定義されたＳＬＡ特性を有する、ストレージクラスの固定組を定義する場合があり、クライアント５０は、特定のオブジェクト３０と特定のストレージクラスを関連付けることを選択してもよい。例えば、初期ストレージクラスは、オブジェクト３０が少なくとも３回、少なくとも２つの異なる領域３１０に複製されるように指定してもよい。高信頼性ストレージクラスは、オブジェクト３０が少なくとも５回、少なくとも３つの異なる領域３１０に複製されるように指定してもよい。予算ストレージクラスは、オブジェクト３０の１つの複製が単一領域３１０に保存されるように指定してもよい。ローカルストレージクラスは、オブジェクト３０が少なくとも３回、単一領域３１０に複製されるように指定してもよい。その他の実施形態において、ストレージサービスシステムは、その他の特徴を有するストレージクラスを定義する、又はクライアント５０が、ストレージポリシー（例えば、ノードピッカー１３０に関して上記に記載されるような）の組み合せを指定することにより、所定のオブジェクト３０のストレージクラスをカスタマイズすることを可能にする。

【０２４７】

上記に記載されるように、ＳＬＡ特性は、複製の数及び複製が分散されるべき領域の数を越えて拡張してもよい。一実施形態において、特定のストレージクラスのＳＬＡ特性は、特定のストレージクラスに関連するオブジェクト３０に対する期待処理待ち時間の指示を含む場合がある。例えば、１つのストレージクラスは、所定の費用で低期待処理待ち時間を指定してもよいが、別のストレージクラスは、より低コストで高期待処理待ち時間を指定してもよい。異なるレベルの期待処理待ち時間が、様々な方法で実装されてもよい。例えば、所定のコーディネータ１２０の観点から、いくつかのノード１６０は、ノード１６０と所定のコーディネータ１２０の近接ノード１６０で利用可能なリソースのレベル及び種類、ノード１６０の処理負荷、又はその他の関連した要因により、他より短いアクセス待ち時間を示すことができる。従って、所定のストレージクラスにより指定されるその他のＳＬＡ特性によりもたらされる制約に従い、一部の実施形態において、コーディネータ１２０及び／又はノードピッカー１３０は、より短い期待処理待ち時間を指定するストレージクラスのオブジェクト３０に対して、より短いアクセス待ち時間を示すノード１６０を選択するように設定されてもよい。その他の実施形態において、コーディネータ１２０は、オブジェクト３０に関連するストレージクラスの期待処理待ち時間に従い、オブジェクト３０に対するクライアントのアクセス要求の処理に優先順位をつけるように設定されてもよい。例えば、コーディネータ１２０は、低期待処理待ち時間を有するストレージクラスを支持し、処理にバイアスをかける一方、高期待処理待ち時間を有するストレージクラスの要求が、最終的には確実に完了するように設定されている、異なるキュー又はその他の処理制御、あるいはデータ構造を実装してもよい。

【０２４８】

ストレージクラスは、オブジェクト３０が最初にストレージサービスシステムに保存される時点で、クライアント５０により指定されてもよい。あるいは、一部の実施形態において、クライアント５０は、オブジェクト３０がストレージサービスシステム内に存在する間のいずれの時点で、オブジェクト３０に関するストレージクラスを変更してもよい。オブジェクト３０が最初に保存される時点で、クライアント５０によりストレージクラスが指定されていない場合、上記に記載されるような初期ストレージクラスが使用される。

上記に記載されるように、一部の実施形態において、オブジェクト３０のストレージクラスは、オブジェクト３０のキーに関連するキーマップ記録１４８内に保存されてもよい。そのような実施形態において、コーディネータ１２０及び／又はレプリケータ１８０は、オブジェクト３０の既存の複製を保存、複製、及び維持する際に、オブジェクト３０のストレージクラスを考慮するように設定されてもよい。クライアント５０は、異なるストレージクラスに関連するオブジェクト３０に対して、異なる使用料を請求される場合があることが考慮される。例えば、高信頼性ストレージクラスは、一般的によりシステムリソースを使用する場合があり、一方、予算ストレージクラスは、より少ないリソースを使用してもよい。従って、所定のサイズのオブジェクト３０において、クライアント５０は、前者のストレージクラスを使用するオブジェクト３０の保存に対してはより多く、後者に対してはより少なく請求される。

【０２４９】

ストレージサービスシステム内のストレージクラスの動作方法の一実施形態を、図２６に図示する。図示される実施形態において、動作は、クライアント５０が特定のオブジェクト３０と関連するストレージクラスを指定するブロック２６００から始まる。続いて、ストレージクラスは、ストレージサービスシステム（ブロック２６０２）内において、特定のオブジェクト３０と永続的に関連付けられる。例えば、ストレージクラスの指示は、クライアント５０の代わりに、コーディネータ１２０によって、特定のオブジェクト３０に関連するデータ構造内、例えばキーマップ記録１４８に保存される。オブジェクト３０に関連するオブジェクトデータの状態は、そして指定されたストレージクラスの特性に従って設定される（ブロック２６０４）。例えば、ストレージクラスが、領域３１０内のオブジェクトの複製の数及び／又は分散に対して特定の要件を指定する場合、コーディネータ１２０及び／又はレプリケータ１８０は、特定のオブジェクト３０に関連するストレージシステムの結果的に生じる状態が、ストレージクラスの要件を満たすように、必要な複製を作成及び分散するように動作する。一部の実施形態において、レプリケータ１８０は、オブジェクト３０に対するストレージクラス要件が長い間確実に維持されるように設定されてもよい。例えば、複製が機能しなくなった場合、レプリケータ１８０は障害を検出し、さらなる複製を作成するように設定されてもよい。

【０２５０】

一部の実施形態において、所定のストレージクラスにより指定されるストレージ特性は、ビットストアノード１６０を介して利用可能な異なる種類のストレージリソースの識別を含む場合がある。例えば、一部の実施形態において、いくつかのビットストアノード１６０は、他より高性能なストレージデバイスを含む、又はそれぞれのビットストアノード１６０は、高性能及び低性能なデバイスの組み合せを含む場合がある。そのような実施形態において、ストレージクラスは、そのクラスに関連するオブジェクト３０に対して、どちらか一方のデバイスを使用するように指定してもよい。

【０２５１】

動的複製

上記に記載されるように、一部の実施形態において、ノードピッカー１３０は、特定のオブジェクト３０の複製が書き込まれるべき特定のビットストアノード１６０を識別する書き込みプランを作成するように設定されてもよい。そのような書き込みプランは、一度書き込みプランが、例えばコーディネータ１２０などによって導入されると、例えば特定のオブジェクト３０に関する様々な書き込みポリシーが満たされるように作成されてもよい。例えば、書き込みプランにより指定されるノード１６０の数は、特定のオブジェクト３０の複製の最小要求数、複製を分散する異なる領域３１０の最小数、又はあらゆるその他のストレージポリシー配慮に従い、決定されてもよい。

【０２５２】

一部の実施形態において、ノードピッカー１３０は、ノード１６０の現行状態を考慮することなく、予測可能な手順に従い、ノード１６０を連続して選択する静的方法にて、書き込みプランを作成するように設定されてもよい。例えば、ノードピッカー１３０は、文

字列複製のためにノード１６０の同一組を連続して選択する、又はラウンドロビン法で沢山のノード１６０を回転してもよい。しかし、大規模な実装において、ストレージサービスシステムは、様々な時点において大幅に異なる状態で動作し得る、多くのノード１６０を含む場合がある。例えば、いくつかのノード１６０は動作不能であり、その他は動作しているが要求アクティビティにより飽和状態であるか、又は使用可能なリソースが少ない、そしてその他は比較的遊休状態又は十分な使用可能リソースを有してもよい。

【０２５３】

さらに、異なるノード１６０は、それぞれの任意のコーディネータ１２０又はノードピッカー１３０から、異なるレベルの通信費用を提示してもよい。例えば、コーディネータ１２０と同一の領域３１０又はデータセンタ３００内に存在するノード１６０は、ローカルで応答時間の短いネットワーク接続を介して、アクセス可能であってもよい。一方、コーディネータ１２０とは異なる領域３１０又はデータセンタ３００に存在するノード１６０は、ローカルノード１６０より大幅に長い待ち時間を示すことができる。さらに、一部の実施形態において、領域３１０間又はデータセンタ３００間の通信は、ローカル通信とは異なる経済費用モデルの通信ネットワークにおいて行われる場合がある。例えば、領域３１０内での通信は、データ転送において、使用量に基づく請求のない、十分なバンド幅を有するプライベートローカルエリアネットワーク（ＬＡＮ）において行われる場合がある。一方、データセンタ３００間の通信は、専用通信設備、公共インターネット、プライベート広域ネットワーク（ＷＡＮ）設備、又はその他の長距離通信ネットワークなどの設備において行われる場合がある。これらの設備は、一般的にＬＡＮ設備よりバンド幅に制限がある場合があり、一部の例において、ＬＡＮ通信には適用されない利用費用（例えば、ピーク又は合計バンド幅使用量に基づく）が第三者により請求されてもよい。

【０２５４】

様々なノード１６０の両方の動作状態及びこれらのノードの通信費用は、長期に渡り、変動してもよい。例えば、ある時点において動作している、又は遊休状態のノード１６０は、後に動作不能又は使用中になる場合があり、またその逆も同様である。同様に、待ち時間及び／又は経済費用などの通信費用は、ある期間において高くなり、その他では低くなる（例えば、ピーク時対ピーク時以外の利用量）場合がある。この変動のため、ある時点において効率的で低コストな書き込みプランは、大幅に効率が低く、高コストである、また別の時点（例えば、書き込みプランに指定されているノード１６０が使用中になる、通信が遅くなる、又は動作不能になる場合）において、実行不可能でさえある場合もある。

【０２５５】

従って、一部の実施形態において、ノードピッカー１３０は、任意のオブジェクト３０の書き込み複製に対する任意の書き込みプランを、ノード１６０の現行状態情報に従い、動的に決定するよう設定されてもよい。一般的に言うと、動的に決定される書き込みプランは、ノード１６０の監視可能な動的状態情報を考慮してもよい。すなわち、動的に決定される書き込みプランは、時間とともに変化するノードの状態情報の機能として作成されてもよい。従って、任意のオブジェクト３０の動的に決定される書き込みプランは、ノード１６０の状態から独立して決定されてもよい、静的に作成される書き込みプランと比較し、ノード１６０の基礎状態情報により、時間とともにそれ自身を変化してもよい。

【０２５６】

上記に記載されるように、多くの異なる種類の状態情報は、書き込みプランの動的作成に考慮されてもよい。一般的に、ノード１６０の状態情報は、任意のノード１６０に関する状態情報、及びそれを介して任意のノード１６０にアクセス可能な通信リソースに関する状態情報（例えば、ネットワークリソース）を含む場合がある。様々な実施形態において、任意のノード１６０に関する状態情報は、任意のノード１６０（又はノードに関連するアプリケーションインスタンス）がＯＫであるかどうか、隔離されているか、又は上記に記載されるようなＤＦＤＤ１１０により示されるその他の動作状態などの、任意のノードの動作状態を含む場合がある。任意のノード１６０に関する状態情報は、動作状態情報

より詳細な任意のノード１６０の行動を示す場合がある負荷状態情報も含む場合がある。例えば、様々な実施形態において、負荷状態情報は、任意のノード１６０に対応するプロセッサ利用レベル、メモリ利用レベル、ストレージデバイス利用レベル、ストレージデバイス入力／出力バンド幅利用レベル、又はネットワークインターフェースのバンド幅利用レベル、又はノード動作のいずれのその他の計測可能な局面を示すことができる。上記に記載されるように、一部の実施形態において、動作状態情報に加え、負荷状態情報は、ＤＦＤＤ１１０を介して入手可能であってもよい。

【０２５７】

ネットワーク費用情報とも称される場合がある通信リソース状態情報は、任意のノード１６０への１つ以上の通信経路の状態に関するいずれの適切な情報を含む場合がある。様々な実施形態において、ネットワーク費用情報は、任意のノード１６０への及び／又はからのメッセージの伝達に関連するネットワーク通信待ち時間を示す場合があり、時間（秒、ミリ秒など）、ネットワークホップ数（例えば、メッセージを伝達するためのルーティングステップ数）、又は別の適切な測定基準により表されてもよい。一実施形態において、ネットワーク費用情報は、任意のノード１６０との通信に利用可能な利用可能バンド幅（例えば、データ転送率）の指示を含む場合がある。別の実施形態において、ネットワーク費用情報は、任意のノード１６０とのネットワーク通信に関係する経済費用の指示を含む場合がある。例えば、そのような費用は、ある量のデータの転送又は受信に対して請求される率、又はいずれのその他のネットワーク通信に適切な費用又は率モデルにより表されてもよい。

【０２５８】

ノードピッカー１３０は、通常、オブジェクト３０の書き込みプランの動的な決定において、ノード１６０の状態情報のいずれの適切な機能を使用してもよい。一部の実施形態において、ノードピッカー１３０により導入されるストレージポリシー（前述のストレージポリシーの追加又は代わりであってもよい）は、特定のオブジェクト３０に対する書き込みプランに含めることが可能である、ノード１６０を制約する状態情報に対するガイドライン又は要件を指定してもよい。様々な実施形態において、これらのポリシーは、オブジェクト３０の特定の組（例えば、特定のストレージクラス又はバケットに含まれており、共通鍵プレフィックスを有する、又は組のメンバーであると表示されている、オブジェクト）に対して全体的に（例えば、すべてのオブジェクト３０に対して）、又は個別のオブジェクト３０（例えば、オブジェクト３０に関する特定のポリシーを指定するクライアントへの応答として）に対して、あるいはいずれの適切なこれらの組み合せで適用されてもよい。一実施例において、特定のストレージクラスは、いくつかの最小数の複製は、いくつかの最大通信待ち時間以下を示すことを要求するストレージポリシーを指定してもよい。同様に、このストレージクラスのオブジェクト３０への書き込みプランの作成において、ノードピッカー１３０は、指定される最大通信待ち時間を満たすかどうかに従い、少なくともいくつかのノード１６０を選択するように設定されてもよい。

【０２５９】

一部の実施形態において、ノードピッカー１３０は、ノード状態情報における様々な種類の最適化に従い、書き込みプランを作成するようにも設定されてもよい。例えば、特定の最大ネットワーク費用又はその他の書き込みプランに関連する費用を指定する代わりに、ストレージポリシーは、特定の時間において利用可能なリソース間で費用が最小限化されることを指定してもよい。同様に、ノードピッカー１３０は、例えば、低ネットワーク通信又はその他の関連費用を有するノード１６０を選択することによって、書き込みプランに関連する１つ以上の費用を最小化するように設定されてもよい。一部の実施形態において、そのような最小化は、その他のノード状態情報要件を指定するその他のストレージポリシーなどの、その他の制約の存在下において起こる場合がある。

【０２６０】

さらに、一部の実施形態において、いくつかのノード状態情報は、予測可能な形で時間とともに変化することに留意されたい。例えば、データセンタ３００間のネットワーク通

信に関連するバンド幅費用は、明確な料金表に従い、変化する。一部の実施形態において、書き込みプランに関連する費用を最小化することは、特定の時間帯に関連する費用に基づき、書き込みプランのすべて又は一部が実行されるべき時間帯を識別することを含む場合がある。例えば、ノードピッカー１３０は、遠隔データセンタ３００と通信するためのバンド幅が、将来のある時点において現時点より安価になることを判断する場合があり、さらに遠隔データセンタ３００に存在するノード１６０を含む書き込みプランの費用は、遠隔データセンタを対象とする少なくともそれらのストレージ作業を、特定の将来時に実行することで最小化できる可能性があることを判断する。このプロセスの１つの起こりうる結果は、ノードピッカー１３０により作成される書き込みプランが、任意のオブジェクト３０のいくつか（又は場合によりすべて）の複製の作成は、特定の将来時まで延期されるべきであるということを示す場合があることである。

【０２６１】

多くの異なるストレージポリシーを特定のオブジェクト３０に適用することは可能である。さらに、ある場合において、特定のオブジェクト３０に関連する各ストレージポリシーを満たす１つの書き込みプランを作成することは不可能である場合がある。例えば、特定のオブジェクト３０に関連するストレージポリシーは、複製の最小数を、最小数の異なる領域３１０に渡り、保存し、配布することを指定してもよい。しかし、オブジェクト３０に対して、書き込みプランが作成された時点では、ノードピッカー１３０が実行中の領域３１０は、一時的通信障害により、その他の領域３１０から一時的に隔離されてもよい。従って、対応するストレージポリシーを満たしつつ、複製をその他の領域３１０に正常に配布することは、少なくとも一時的に不可能である場合がある。

【０２６２】

一実施形態において、ノードピッカー１３０は、オブジェクト３０に対する書き込みプランを、書き込みプランが満たすストレージポリシーの数を最大化するという原則において、動的に決定するように設定されてもよい。次善条件の存在下において、これは、ストレージポリシーを満たすために「最善努力」を示す書き込みプランをもたらす場合がある。例えば、今記載した特定のシナリオにおいて、通信障害のため、領域別ポリシーは満たせない場合があるが、最小複製ポリシーは、特定のオブジェクト３１０の要求される最小数の複製を、ローカル領域３１０内に保存することで、満たされてもよい。一部の実施形態において、ストレージポリシーの最大化は、様々な抑制化で行われる場合がある。例えば、それらが満たされない場合、書き込みプランは決定されず、クライアントのオブジェクトを保存するという要求は、失敗する可能性があるというような、いくつかのストレージポリシーは、義務として認識されてもよい。その他のストレージポリシーは、選好関係又は重み付けを有する場合があり、例えば、最大化プロセスにおいて、低い選好ストレージポリシーの中から、より高い選好ストレージポリシーが選択されてもよい。別の実施形態において、ストレージポリシーの選択は、結果として生じるストレージプランにより満たされるストレージポリシーの数の代わりに、又はそれへの追加として、結果として生じるストレージプランの合計重み（満たされるストレージポリシーの重みに基づき決定される）を最大化することにより実行されてもよい。

【０２６３】

オブジェクト３０に対する書き込みプランを動的に決定するための様々な技術は、オブジェクト３０が最初に保存される際、単独で実行する必要はないことに留意されたい。上記に記載されるように、一部の実施形態において、レプリケータ１８０は、オブジェクト３０の複製がアクセス可能であるかどうかを決定するために、オブジェクト３０に対応するキーマップエントリ１４４を調査するように設定されてもよい。特定のオブジェクト３０のいずれの複製もアクセス可能でない場合、レプリケータ１８０は、さらなる複製を作成するために使用されてもよい、新しい書き込みプランをノードピッカー１３０から要求するように設定されてもよい。新しい書き込みプランは、上記に記載される技術のいずれの適切な組み合せを使用するノードピッカー１３０により、動的に決定されてもよい。さらに、一部の実施形態において、レプリケータ１８０は、オブジェクト３０の様々なスト

レージポリシーへの順守をより広く監視するように設定されてもよい。例えば、レプリケータ１８０は、オブジェクト３０の既存の複製の組が、最小複製ポリシー、又はいずれのその他の適切なポリシーの組に加え、領域別ポリシーを満たすかどうかを判断するように設定されてもよい。そのような一実施形態において、レプリケータ１８０が、特定のオブジェクト３０の既存の複製により満たされているポリシーの数が、基準値より少ないと判断する場合、レプリケータ１８０は、上記に記載されるように、満たされるストレージポリシーが最大になるように動的に決定される新しいストレージプランをノードピッカー１３０から要求してもよい。別の実施形態において、レプリケータ１８０は、特定の義務ストレージポリシーが満たされないという判断に伴い、又は満たされるストレージポリシーの合計重みが基準値を下回るという判断に伴い、新しいストレージプランを要求してもよい。

【０２６４】

図２７は、ビットストアノード１６０の現行状態情報に従い、データオブジェクトの１つ以上の複製を保存するための書き込みプランを動的に決定する方法の一実施形態を図示する。図示される実施形態において、操作は、任意のオブジェクト３０を保存するというクライアントの要求が受け取られるブロック２７００から始まる。一実施形態において、そのような要求は、上記に詳細が記載されるウェブサービスインターフェース１００を介して、ウェブサービスプロトコルに従い受け取られる場合がある。

【０２６５】

続いて、ビットストアノード１６０の現行状態情報に従い、任意のオブジェクト３０の複製を保存するための書き込みプランが動的に決定される（ブロック２７０２）。例えば、ノードピッカー１３０は、任意のオブジェクト３０に適用されてもよい様々なストレージポリシーに従い、書き込みプランを決定するように設定されている場合があり、当該ポリシーは、ノードの動作状態、ノードの負荷状態情報、ネットワーク通信費用、又は上記に詳細に記載されるような、いずれのその他の適切な状態情報などの、いずれの適切な現行状態情報を考慮する。さらに、上記に記載されるように、一部の実施形態において、動的な書き込みプランの決定は、書き込みプランに関連する費用を最小化すること、又は書き込みプランにより満たされるストレージポリシーの重み又は数を最大化することなどによる、状態情報又はストレージポリシーに関する最適化を含む場合がある。

【０２６６】

任意のオブジェクト３０の複製は、そして動的に決定された書き込みプランに従い、１つ以上のビットストアノード１６０に保存される（ブロック２７０４）。例えば、コーディネータ１２０は、上記に記載されるように、書き込みプランに指定されるそれぞれのビットストアノード１６０に対する操作を設置するビットストアオブジェクトを作成するように設定されてもよい。一部の実施形態において、書き込みプランのいくつかのストレージ操作は、上記に記載されるように、その他の操作とは異なる時間に実行されてもよい。

【０２６７】

前述のように、一部の実施形態において、書き込みプランは、１つ以上の複製がビットストアノード１６０に渡り既に保存されているオブジェクト３０に対して動的に決定されてもよい。図２８は、そのような方法の一実施形態を図示する。図示される実施形態において、操作は、任意のオブジェクト３０の１つ以上の既存の複製が調査されるブロック２８００から始まる。例えば、上記に記載されるように、レプリケータ１８０の一実施形態は、任意のオブジェクト３０の既存の複製がアクセス可能であるか、及び／又は任意のオブジェクト３０の既存の複製がどの程度オブジェクトに関連するストレージポリシーを満たすかどうかを判断するように設定されてもよい。

【０２６８】

任意のオブジェクト３０の複製の調査を受けて、１つ以上のさらなる複製を作成する必要があると判断されてもよい（ブロック２８０２）。例えば、既存の複製は、故障しているか、又は別の理由でアスセス不可となり、複製の最小数を下回っている場合がある。あるいは、既存の複製の状態は、１つ以上のストレージポリシーに対して不十分である場合

がある。続いて、任意のオブジェクト３０のさらなる複製を保存するための書き込みプランは、ビットストアノード１６０の現行状態情報に従い、動的に決定される（ブロック２８０４）。そのような書き込みプランは、前述と同様に、又はいずれの適切なその変更に従い、決定されてもよい。一部の実施形態において、任意のオブジェクト３０のさらなる複製を作成する必要がないと判断される場合、書き込みプランは決定されない場合があることに留意されたい。

【０２６９】

任意のオブジェクト３０の複製は、そして動的に決定された書き込みプラン（ブロック２８０６）に従い、１つ以上のビットストアノード１６０に保存される。例えば、レプリケータ１８０は、上記に記載されるように、書き込みプランに指定されるそれぞれのビットストアノード１６０に対して操作を設置するビットストアオブジェクトを作成するように設定されているか、又は単に任意のオブジェクト３０の既存の複製を保存する１つ以上のノード１６０を、書き込みプランに指定されるノード１６０の１つ以上に直接それらの複製をコピーしてもよい。

【０２７０】

例示的なコンピュータシステムの実施形態

一部の実施形態において、上記に記載されるいずれの方法又は技術は、プログラムのインストラクション及びコンピュータアクセス可能な媒体を介して保存、又は伝達することが可能なデータとして実装されてもよい。そのような方法又は技術は、例えば、ストレージクライアント５０、ウェブサービスプラットフォーム１００、ＤＦＤＤ１１０、コーディネータ１２０、ノードピッカー１３０、キーマップインスタンス１４０、ビットストアノード１６０、レプリケータ１８０、及び／又はレプリケータキーマップ１９０を含む場合があるが、制限はない。そのような方法又は技術は、図６～９、１３～１５、２０～２２、及び２５～２８に図示されているいずれの方法ならびにその適切な変更をさらに含む場合がある。そのようなプログラムのインストラクションは、上記に記載される特定の方法又は一部の方法などの特定のコンピュータ機能を実行し、より一般的なオペレーティングシステムの機能性、アプリケーションの機能性、及び／又はいずれのその他の適切な機能を提供するために、実行されてもよい。一部の実施形態において、上記に記載されるコンポーネント又は方法は、異なるその他の実施形態において、これらの示されているものより少ないエンティティに組み込まれる、又は機能性はコンポーネント又は方法に渡り、上記に記載される分割とは異なる形で分割されてもよい。

【０２７１】

コンピュータアクセス可能な媒体を含むコンピュータシステムの例示的な一実施形態を図２９に図示する。そのようなシステムは、ノードとも称される場合がある。前述のように、一実施形態において、上記に記載されるいずれの様々なストレージシステムコンポーネントは、多くのノードを渡り分散されている場合があり、そのような任意のコンポーネントは、１つ以上のノードにより実装されている、又はいくつかのノードに渡り分割されてもよい。一部の実施形態において、ノードは単一ストレージサービスシステムコンポーネントの機能を排他的に実装してもよいが、その他の実施形態において、ノードは、いくつかの異なるシステムコンポーネントのすべて、又は部分的な機能性を実装してもよい。図示される実施形態において、コンピュータシステム２９００は、入力／出力（Ｉ／Ｏ）インターフェース２９３０を介してシステムメモリ２９２０と対になっている１つ以上のプロセッサ２９１０を含む。コンピュータシステム２９００は、Ｉ／Ｏインターフェース２９３０と対になっているネットワークインターフェース２９４０をさらに含む。

【０２７２】

様々な実施形態において、コンピュータシステム２９００は、１つのプロセッサ２９１０を含むユニプロセッサシステム、又はいくつかのプロセッサ２９１０（例えば、２つ、４つ、８つ、又は別の適切な数）を含むマルチプロセッサシステムであってもよい。プロセッサ２９１０は、命令を実行する能力を有する、任意の適切なプロセッサであってもよい。例えば、様々な実施形態において、プロセッサ２９１０は、ｘ８６、ＰｏｗｅｒＰＣ

、ＳＰＡＲＣ、又はＭＩＰＳ　ＩＳＡ、あるいはその他の適切なＩＳＡなどの様々な命令組アーキテクチャ（ＩＳＡ）を実装する汎用、又は内蔵プロセッサであってもよい。マルチプロセッサシステムにおいて、それぞれのプロセッサ２９１０は、必ずではないが、通常、同一ＩＳＡを実装する。

【０２７３】

システムメモリ２９２０は、プロセッサ２９１０によりアクセス可能に命令及びデータを保存するように設定されてもよい。様々な実施形態において、システムメモリ２９２０は、静的ランダムアクセスメモリ（ＳＲＡＭ）、同期動的ＲＡＭ（ＳＤＲＡＭ）、非揮発性／フラッシュ型メモリ、又はいずれのその他の種類のメモリなど、いずれの適切なメモリ技術を使用し、実装されてもよい。図示される実施形態において、上記に詳細を記載する、いずれのこれらのストレージサービスシステムコンポーネント及びその他の機能などの所望の機能を実装するプログラムのインストラクション及びデータは、システムメモリ２９２０内にコード２９２５として保存されることを示す。

【０２７４】

一実施形態においてＩ／Ｏインターフェース２９３０は、ネットワークインターフェース２９４０又はその他の周辺インターフェースを含む、プロセッサ２９１０、システムメモリ２９２０、及びデバイス内のいずれの周辺装置間のＩ／Ｏトラフィックを調整するように設定されてもよい。一部の実施形態において、Ｉ／Ｏインターフェース２９３０は、いずれの必要なプロトコル、タイミング調節、又はその他のあるコンポーネント（例えば、システムメモリ２９２０）からのデータ信号を別のコンポーネント（例えば、プロセッサ２９１０）での使用に適切なフォーマットに変換するためのデータ変換を実行してもよい。一部の実施形態において、Ｉ／Ｏインターフェース２９３０は、例えば、周辺装置相互接続（ＰＣＩ）バス標準又はユニバーサルシリアルバス（ＵＳＢ）標準の異型などの様々な種類の周辺装置のバスを通じて接続されたデバイスのサポートを含む場合がある。一部の実施形態において、Ｉ／Ｏインターフェース２９３０の機能は、例えば、ノースブリッジ及びサウスブリッジなど、２つ以上の分離したコンポーネントに分割されてもよい。また、一部の実施形態において、システムメモリ２９２０に対するインターフェースなどの、Ｉ／Ｏインターフェース２９３０の機能性の一部又はすべては、プロセッサ２９１０に直接組み込まれている場合がある。

【０２７５】

ネットワークインターフェース２９４０は、コンピュータシステム２９００とネットワークに接続されたその他のデバイス、例えばその他のコンピュータシステムなどとのデータ交換が可能になるように設定されてもよい。様々な実施形態において、ネットワークインターフェース２９４０は、有線又は無線の一般的なデータネットワーク、例えば、イーサネット（登録商標）ワークのいずれの適切な種類、アナログ音声ネットワーク又はデジタルファイバー通信ネットワークなどの電気通信／電話ネットワーク、ファイバーチャネルＳＡＮなどのストレージ領域ネットワーク、又はその他のいずれの適切な種類のネットワーク又はプロトコルなどを介して、通信をサポートしてもよい。

【０２７６】

一部の実施形態において、システムメモリ２９２０は、上記に記載されるように、プログラムのインストラクション及びデータを保存するように設定されたコンピュータアクセス可能媒体の一実施形態であってもよい。しかし、その他の実施形態において、プログラムのインストラクション及び／又はデータは、異なる種類のコンピュータアクセス可能媒体から受け取る、へ送る、上に保存してもよい。一般的に、コンピュータアクセス可能媒体は、例えば、Ｉ／Ｏインターフェース２９３０を介してコンピュータシステム２９００と対になっているディスク又はＣＤ／ＤＶＤ－ＲＯＭなどの磁気又は光学媒体などのストレージ媒体あるいはメモリ媒体を含む場合がある。またコンピュータアクセス可能媒体は、コンピュータシステム２９００の一部の実施形態に、システムメモリ２９２０又は別の種類のメモリとして含まれている場合がある、ＲＡＭ（例えばＳＤＲＡＭ、ＤＤＲ　ＳＤＲＡＭ、ＲＤＲＡＭ、ＳＲＡＭなど）、ＲＯＭ、などのいずれの揮発性又は非揮発性媒体

を含む場合もある。コンピュータアクセス可能媒体を介して保存されたプログラムのインストラクション及びデータは、転送媒体、又は電気、電磁気、又はネットワーク及び／又はネットワークインターフェース２９４０を介して実装されている無線リンクなどの通信媒体を介して伝送される場合があるデジタル信号などの信号により、転送されてもよい。

【０２７７】

上記の実施形態は、相当な詳細を記載したが、多数の変更及び修正は、一度上記の開示が完全に理解されると、当技術分野に精通する者により明らかになるであろう。以下の特許請求の範囲は、すべてのそのような変更及び修正を包含することが意図される。

【図面の簡単な説明】

【０２７８】

【図１】ウェブサービスとしてユーザにストレージを提示するためのストレージモデルの一実施形態を示すブロック図である。

【図２】ストレージサービスシステムのアーキテクチャの一実施形態を示すブロック図である。

【図３】ストレージシステムコンポーネントの物理的配置の一実施形態を示すブロック図である。

【図４】ストレージノードの一実施形態を示すブロック図である。

【図５】ストレージノード内のデータオブジェクトを体系化するように構成されるデータストラクチャの一実施形態を示すブロック図である。

【図６】オブジェクト取得操作を実行する方法の一実施形態を示す工程図である。

【図７】オブジェクト格納操作を実行する方法の一実施形態を示す工程図である。

【図８】オブジェクト解放操作を実行する方法の一実施形態を示す工程図である。

【図９】オブジェクトストレージ空間を再圧縮する方法の一実施形態を示す工程図である。

【図１０】キーマップインスタンスデータ構造の一組の一実施形態を示すブロック図である。

【図１１Ａ】キーマップインスタンスの階層的実装の一実施形態を示す。

【図１１Ｂ】キーマップインスタンスの階層的実装の一実施形態を示す。

【図１１Ｃ】キーマップインスタンスの階層的実装の一実施形態を示す。

【図１１Ｄ】キーマップインスタンスの階層的実装の一実施形態を示す。

【図１２】キーマップインスタンス内の階層的層間における関係を要約するブロック図である。

【図１３】キーマップエントリ格納操作を実行する方法の一実施形態を示す工程図である。

【図１４】キーマップエントリが取得操作を実行する方法の一実施形態を示す工程図である。

【図１５Ａ】更新伝播を使用してキーマップインスタンスを同期化する方法の一実施形態を示す工程図である。

【図１５Ｂ】アンチエントロピープロトコルを使用してキーマップインスタンスを同期化する方法の一実施形態を示す工程図である。

【図１６】レプリケータキーマップエントリの一実施形態を示すブロック図である。

【図１７】不均衡インデックスデータ構造の一実施形態を示す。

【図１８】不均衡データ構造において使用するためのインデックスノードの一実施形態を示す。

【図１９】層別インデックスデータ構造の一実施形態を示す。

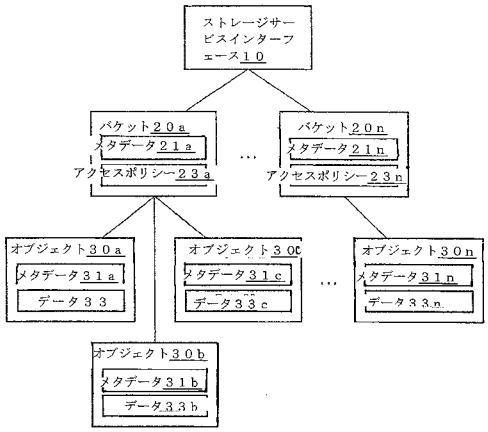
【図２０】不均衡インデックスデータ構造をトラバースするための方法の一実施形態を示す工程図である。

【図２１】ＦＩＮＧＥＲＰＲＩＮＴアンチエントロピープロトコルメッセージを処理する方法の一実施形態を示す工程図である。

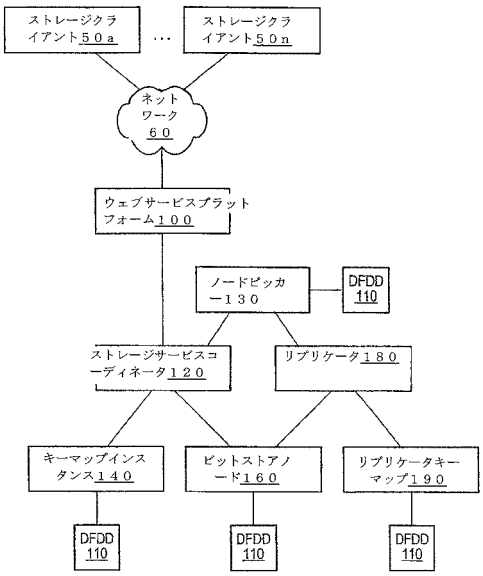
【図２２】ＦＩＬＴＥＲアンチエントロピープロトコルメッセージを処理するための方法

の一実施形態を示す工程図である。
【図２３】発見及び障害検出デーモン（ＤＦＤＤ）の一実施形態を示す。
【図２４】ＤＦＤＤインスタンスによって維持してもよい大域操作状態マシンの一実施形態を示す。
【図２５】ゴシッププロトコルに従ったＤＦＤＤインスタンスを同期化するための方法の一実施形態を示す工程図である。
【図２６】ストレージサービスシステム内のストレージクラスの操作方法の一実施形態を示す工程図である。
【図２７】ストレージノードの現在の情報に従い、データオブジェクトの１つ以上の複製を保存するための書き込みプランを動的に決定する方法の一実施形態を示す工程図である。
【図２８】１つ以上の複製が既にストレージノード間に保存されているオブジェクトに関係する書き込みプランを動的に決定する一実施形態を示す工程図である。
【図２９】コンピュータシステムの例示的な実施形態を示す工程図である。

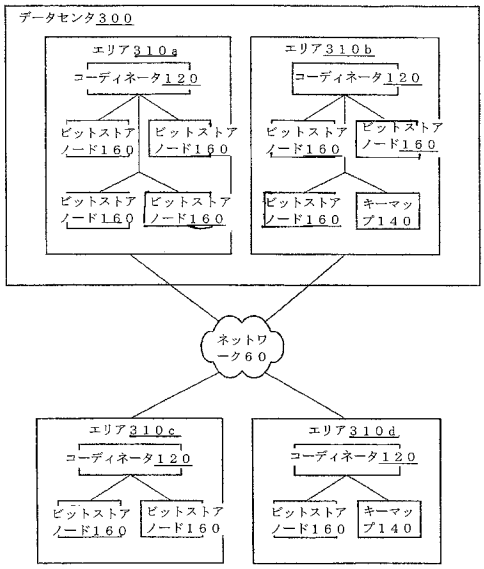
【図１】



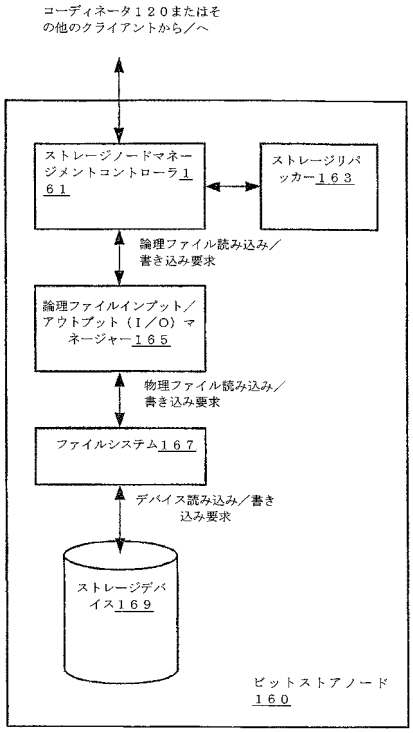
【図２】



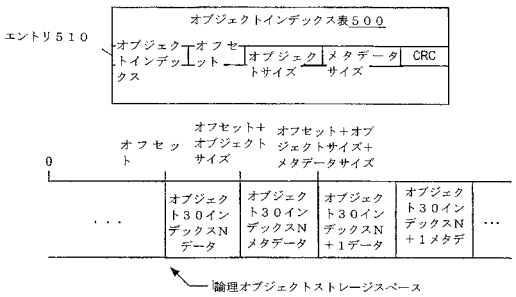
【図３】


データセンタ３００
エリア３１０ａ
コーディネータ１２０
ビットストアノード１６０
ビットストアノード１６０
ビットストアノード１６０
ビットストアノード１６０
エリア３１０ｂ
コーディネータ１２０
ビットストアノード１６０
ビットストアノード１６０
ビットストアノード１６０
キーマップ１４０
ネットワーク６０
エリア３１０ｃ
コーディネータ１２０
ビットストアノード１６０
ビットストアノード１６０
エリア３１０ｄ
コーディネータ１２０
ビットストアノード１６０
キーマップ１４０

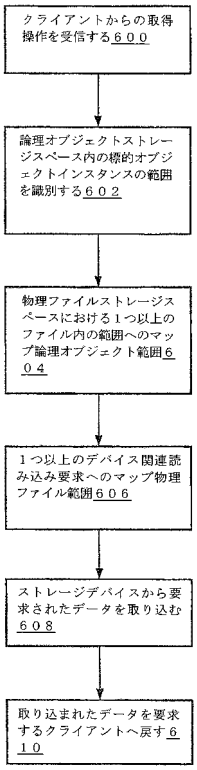
【図４】


コーディネータ１２０またはその他のクライアントから／へ
ストレージノードマネージメントコントローラ１６１
ストレージリパッカー１６３
論理ファイル読み込み／書き込み要求
論理ファイルインプット／アウトプット（Ｉ／Ｏ）マネージャー１６５
物理ファイル読み込み／書き込み要求
ファイルシステム１６７
デバイス読み込み／書き込み要求
ストレージデバイス１６９
ビットストアノード１６０

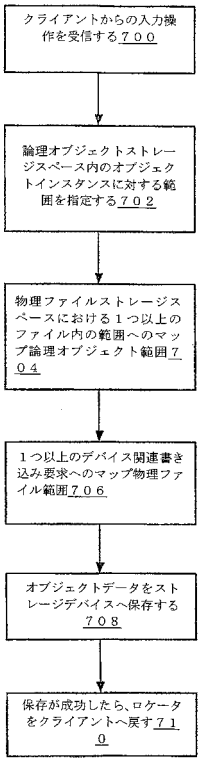
【図５】


エントリ５１０
オブジェクトインデックス表５００
オブジェクトインデックス
オフセット
オブジェクトサイズ
メタデータサイズ
CRC
0
オフセット
オフセット＋オブジェクトサイズ
オフセット＋オブジェクトサイズ＋メタデータサイズ
オブジェクト３０インデックスＮデータ
オブジェクト３０インデックスＮメタデータ
オブジェクト３０インデックスＮ＋１データ
オブジェクト３０インデックスＮ＋１メタデ
論理オブジェクトストレージスペース

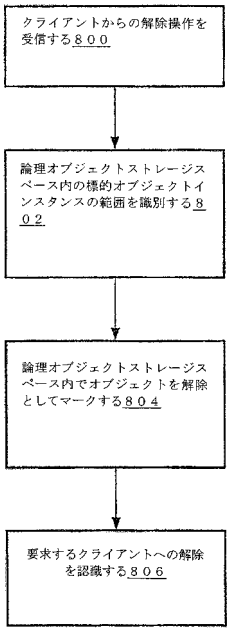
【図６】


クライアントからの取得操作を受信する６００
論理オブジェクトストレージスペース内の標的オブジェクトインスタンスの範囲を識別する６０２
物理ファイルストレージスペースにおける１つ以上のファイル内の範囲へのマップ論理オブジェクト範囲６０４
１つ以上のデバイス関連読み込み要求へのマップ物理ファイル範囲６０６
ストレージデバイスから要求されたデータを取り込む６０８
取り込まれたデータを要求するクライアントへ戻す６１０

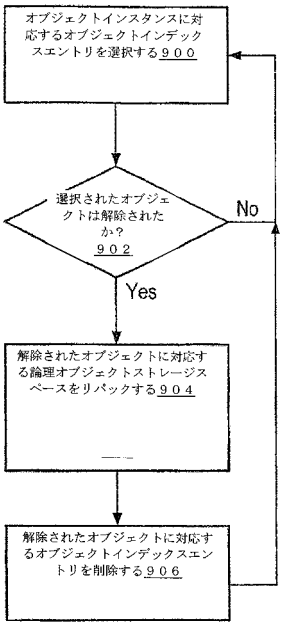
【図７】



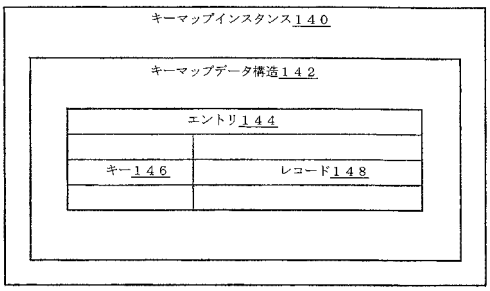
【図８】



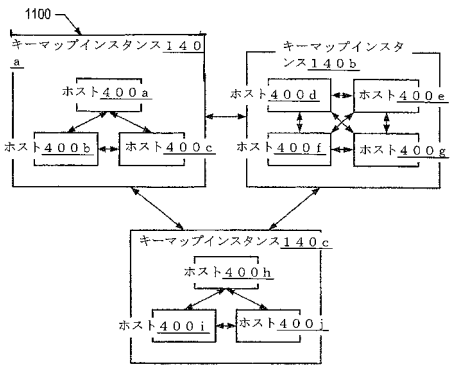
【図９】



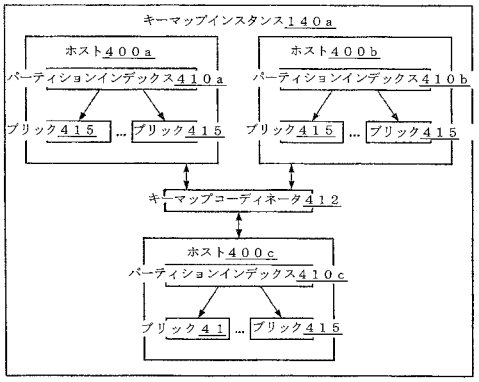
【図１０】



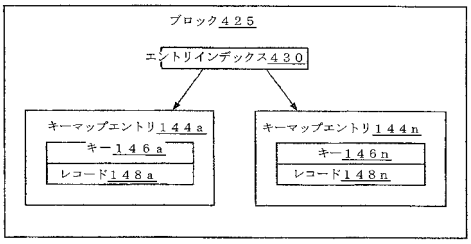
【図１１Ａ】



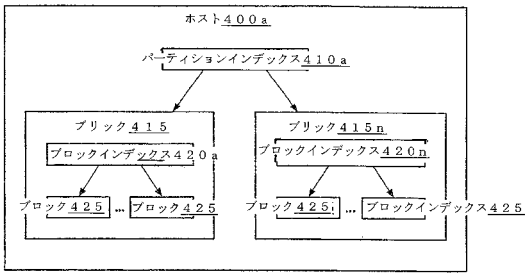
【図１１Ｂ】



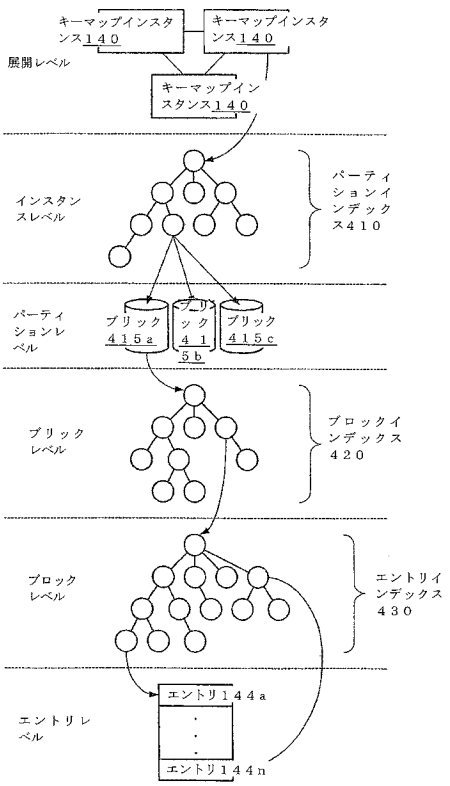
【図１１Ｄ】



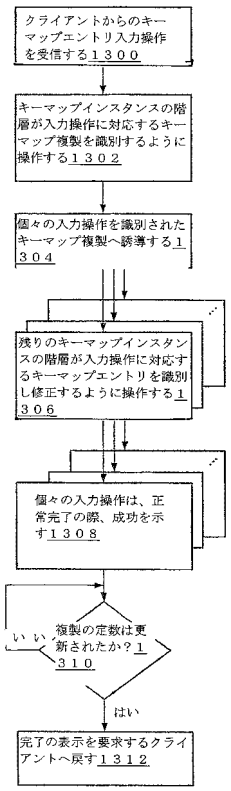
【図１１Ｃ】



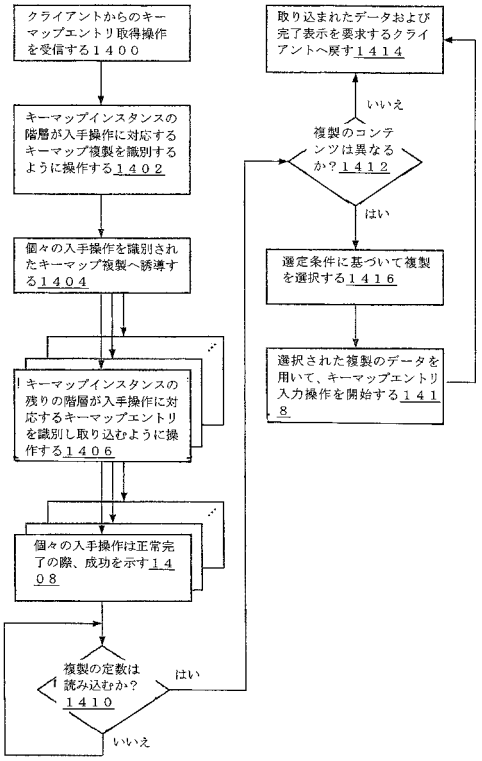
【図１２】



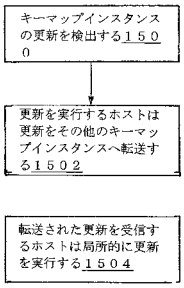
【図１３】



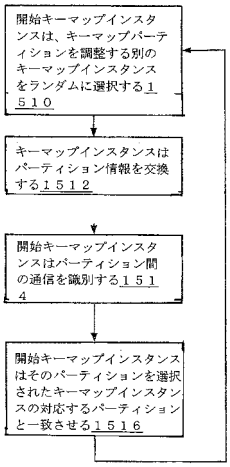
【図１４】



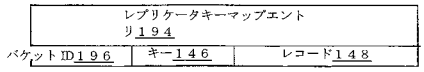
【図１５Ａ】



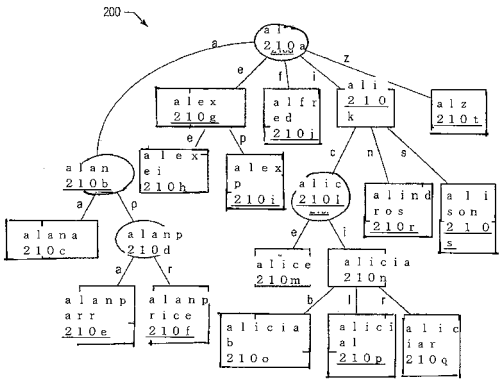
【図１５Ｂ】



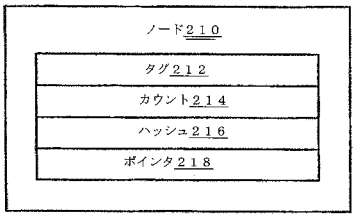
【図１６】



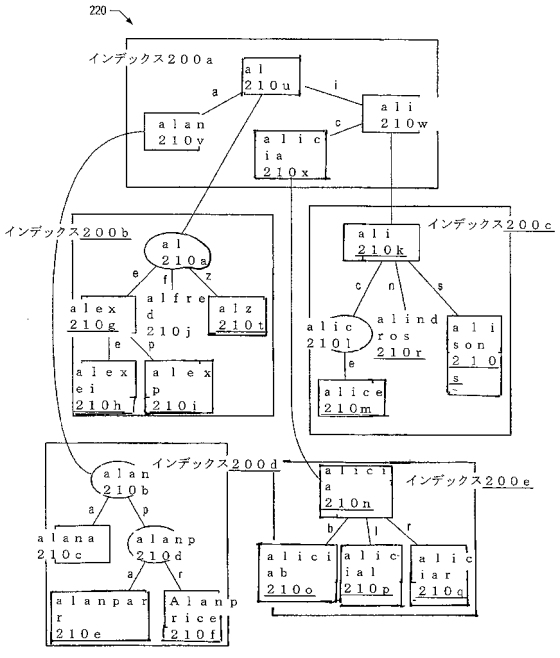
【図１７】



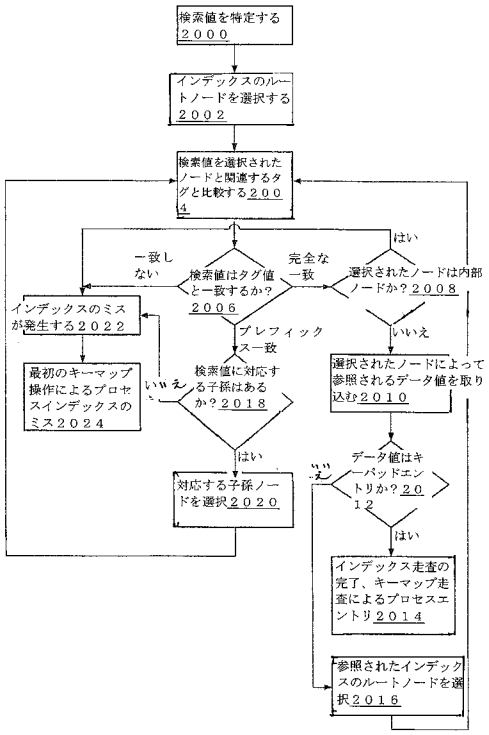
【図１８】



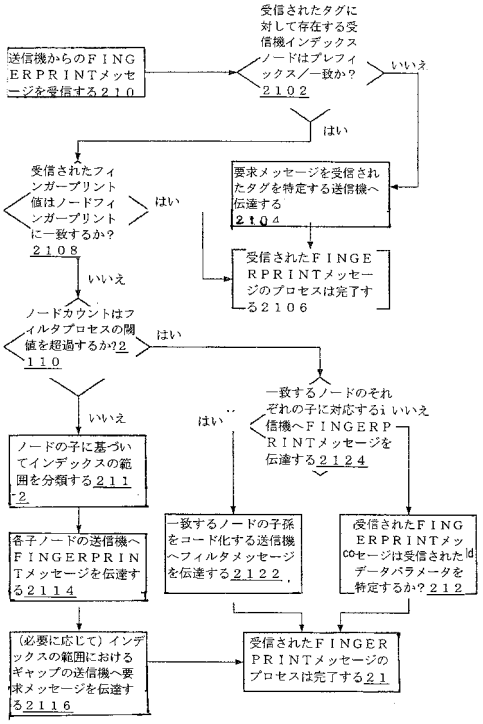
【図１９】



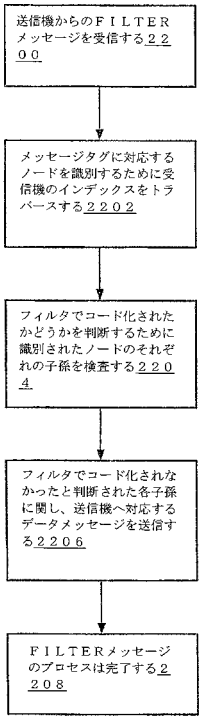
【図２０】



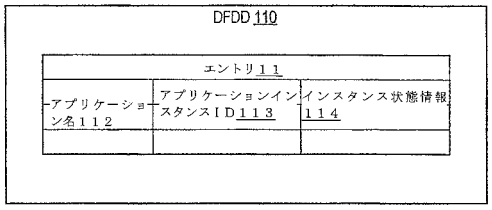
【図２１】



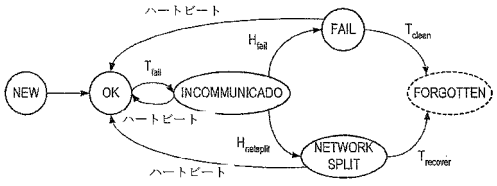
【図２２】



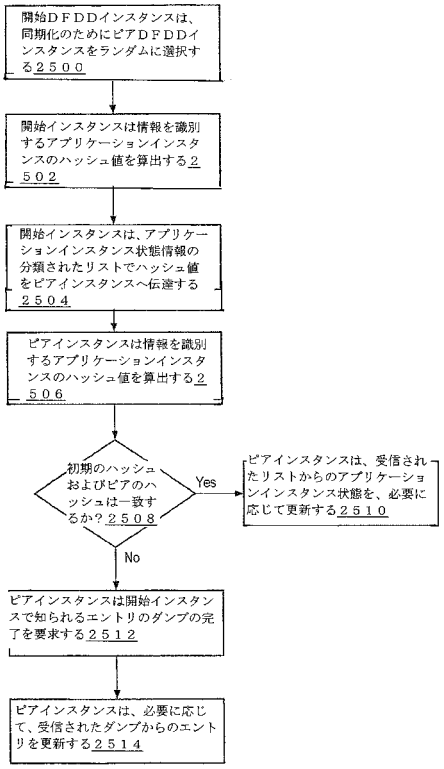
【図２３】



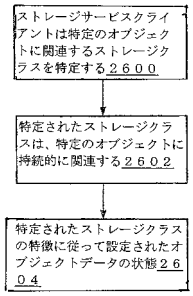
【図２４】



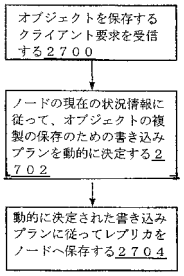
【図２５】



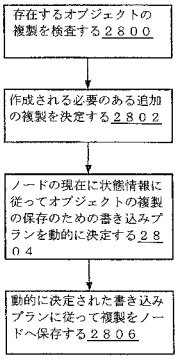
【図２６】



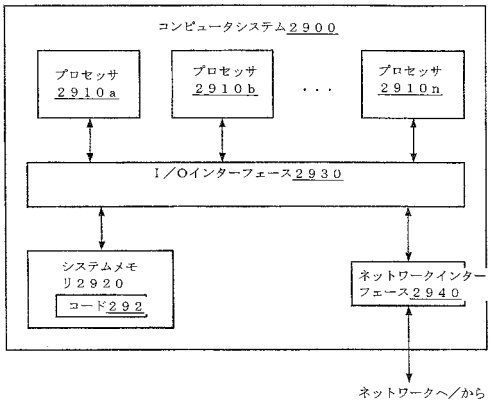
【図２７】



【図２８】



【図２９】



フロントページの続き

(72)発明者 アトラス, アラン・ビイ
アメリカ合衆国・９８１４４・ワシントン州・シアトル・１２ティエイチ アベニュ サウス・１２００・スイート １２００
(72)発明者 バース, デイビッド・エム
アメリカ合衆国・９８１４４・ワシントン州・シアトル・１２ティエイチ アベニュ サウス・１２００・スイート １２００
(72)発明者 コーミー, ジョン・デイビッド
アメリカ合衆国・９８１４４・ワシントン州・シアトル・１２ティエイチ アベニュ サウス・１２００・スイート １２００
(72)発明者 フィッシュマン, エイミ・ケイ
アメリカ合衆国・９８１４４・ワシントン州・シアトル・１２ティエイチ アベニュ サウス・１２００・スイート １２００
(72)発明者 ソレンソン, ジェイムズ・クリストファ・サード
アメリカ合衆国・９８１４４・ワシントン州・シアトル・１２ティエイチ アベニュ サウス・１２００・スイート １２００
(72)発明者 ワグナー, エリック・エム
アメリカ合衆国・９８１４４・ワシントン州・シアトル・１２ティエイチ アベニュ サウス・１２００・スイート １２００

審査官 桜井 茂行

(56)参考文献 国際公開第００／０２６７８２（WO, Ａ１）
特開２００２－０５５８６９（ＪＰ, Ａ）
欧州特許出願公開第０１１６０６９２（ＥＰ, Ａ１）

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 12/00
G06F 17/30
G06F 3/06